Dynamic Puncturable Encryption

Priyanka Dutta, Willy Susilo, Fuchun Guo, and Dung Hoang Duong

Institute of Cybersecurity and Cryptology
School of Computing and Information Technology
University of Wollongong
Northfields Avenue, Wollongong NSW 2522, Australia
{pdutta,wsusilo,fuchun,hduong}@uow.edu.au

Abstract. Dynamic access control is a central requirement in modern encrypted data storage, where decryption privileges must evolve as credentials and permissions change over time. In practice, such changes are routine: users join or leave projects, roles shift, devices are replaced or lost, keys expire or are rotated, and security incidents may necessitate immediate revocation or later restoration of access. Motivated by the need for dynamic access control in encrypted data storage, we explore how to implement puncturable encryption (PE) in environments where user credentials and permissions evolve over time. PE offers a powerful fine-grained revocation functionality. By "puncturing" a secret key, one can revoke decryption capability for specific ciphertexts, enabling fine-grained access control over who can get access to which data. This capability has driven strong theoretical and practical interest since the introduction of PE. However, the existing PE and its variants fall short in dynamic environments where revocation is not always permanent. Specifically, many deployments require the ability to restore access, or to lawfully delegate decryption rights for specific ciphertexts that were previously revoked.

In this work, we bridge this gap by introducing a novel primitive Dynamic Puncturable Encryption (DPE). By "dynamicity", we mean that the system simultaneously supports user-controlled revocation of decryption, trusted authority-controlled revocation or granting of access to data, and delegation of decryption rights based on requirements, all at a finer level. Such a feature further broadens the practical applicability of the PE paradigm and allows dynamic access control. We formalize the concept of DPE, and realize it through a lattice-based construction in the standard model that ensures quantum safety.

Keywords: Updatable Ciphertext · Access Control · Revoke or Restore Decryption Capability · Learning with Errors

1 Introduction

Encryption has become the default mechanism for securing modern data storage systems. It prevents unintended disclosure of data while allowing intended users to obtain decryption capabilities to access it. In practice, decryption eligibility is governed by a trusted authority, which may be a data owner, a security

administrator, or an organizational controller responsible for defining and enforcing policies or restrictions. However, such decryption rights are inherently dynamic rather than static. In real deployments, user accounts and credentials must be provisioned, updated, suspended, and revoked as individuals join or leave projects, change roles, replace or lose devices, or security incidents may require emergency revocation of access. Further, many deployments also require the ability to restore access or to lawfully delegate decryption rights for specific ciphertexts that were previously revoked. NIST [1] explicitly emphasizes these, specifically restricting access when individuals are transferred or terminated and administering accounts in a dynamic, ongoing manner. It creates a practical demand for dynamic access control over encrypted data.

In terms of access control, puncturable encryption (PE) [16], offers a compelling primitive for fine-grained revocation of decryption capabilities. By "puncturing" a secret key with a tag, one can disable decryption of ciphertexts associated with that tag, yielding a fine-grained mechanism to revoke access to selected messages without any interaction with the sender. This core functionality has proven essential for enabling forward secrecy in practical applications, most notably as a building block for forward-secret 0-RTT key exchange [17,8], a critical component in modern TLS protocols [17], secure group messaging [23], cloud email [26], secure deletion [16], etc. To broaden the applications of PE in the distributed storage systems, Derler et al. [9,11] introduced fully PE and dual-form PE, respectively. Additionally, to remove certificate repositories, Wei *et al.* [26] introduced puncturable identity-based encryption (PIBE), in which a secret key corresponding to each identity can be extracted from the master secret key, and recipients may puncture the extracted keys arbitrarily many times. Derler *et al.* [10] later formulated identity-based puncturable encryption, extending PIBE, where secret keys can be extracted not only from the master secret key but also from previously punctured secret keys. Additionally, key puncturing and key extraction can occur in any order. Further, hierarchical identity-based puncturable encryption [12], puncturable attribute-based encryption [22,14] have been introduced. To broaden the applicability of PE, puncturable proxy re-encryption [23] and puncturable identity-based proxy re-encryption [18] combine the functionalities of proxy re-encryption and PE, enabling delegation of decryption rights. However, neither approach provides a mechanism to regain decryption rights for specific ciphertexts after those rights have been revoked through puncturing.

The above limitations of variants of PE create a significant research gap. However, addressing this gap is challenging. The main hindrance to crafting this solution arises from the fundamental conflict between the fine-grained forward secrecy offered by PE and the restoration of decryption capabilities over the same encrypted data. These two features are inherently at odds because the presence of a restoration mechanism could be exploited by an attacker, thereby compromising fine-grained forward secrecy. This dichotomy leads to the following challenging question: "*Can we construct a new primitive in the paradigm of* PE

[1] `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf`

for multi-user systems that supports dynamic access control for users without compromising fine-grained forward secrecy?"

1.1 Our Contributions and Technical Overview

In this work, we answer the above question assertively by introducing Dynamic Puncturable Encryption (DPE), a novel primitive that simultaneously supports dynamic access control and provides fine-grained forward secrecy for users. Specifically, it enables controlled revocation or granting of re-access over encrypted data, and lawful delegation of decryption rights based on requirements, controlled by a trusted authority. It also supports user-controlled revocation of decryption rights at a finer level, while ensuring fine-grained forward secrecy, thereby significantly enhancing prior ideas on PE. We provide a qualitative comparison of DPE with PE schemes and their variants in Table 1.

Table 1. Qualitative Comparison with DPE

Schemes	Revocation of decryption capability	Delegation of decryption rights	Restoration of decryption capability	Dynamic Access Control
PE [16]	✓	×	×	×
PIBE, IBPE[14,13,10]	✓	×	×	×
FuPE, DFPE[9,11]	✓	×	×	×
PPRE[23]	✓	✓	×	×
PIBPRE[18]	✓	✓	×	×
DPE (Section 4)	✓	✓	✓	✓

PE: Puncturable Encryption; PIBE: Puncturable Identity-based Encryption; IBPE: Identity-based Puncturable Encryption; FuPE: Fully Puncturable Encryption; DFPE: Dual-form Puncturable Encryption; PPRE: Puncturable Proxy Re-encryption; PIBPRE: Puncturable Identity-based Proxy Re-encryption; DPE: Dynamic Puncturable Encryption.

To formalize DPE, we introduce a *two-layer revocation mechanism* as one of the main technical ingredients. In this approach, decryption rights are revoked through two coordinated layers of key puncturing under the control of a trusted authority, enabling selective access restoration without undermining the overall system's forward secrecy. We further introduce *dynamic tag updates* to enhance these controls by allowing ciphertexts to be re-tagged, without altering the underlying message. These enable the restoration or lawful delegation of decryption rights through information (tokens) issued by the trusted authority. In a nutshell, we define the syntax and security of DPE, and to realize it, we provide a novel construction from standard lattices that ensures quantum safety.

Formalization of Dynamic Puncturable Encryption.
Formalizing DPE is subtle because it must reconcile two goals that naturally pull in opposite directions: fine-grained forward secrecy and controlled restoration of

decryption capability. The technical challenge is then to let access evolve across both dimensions without creating loopholes that undo puncturing. To capture this dynamicity, we divide the lifetime of a DPE system into a polynomial number of discrete time periods and endow ciphertexts with a time period, and two types of tags, positive and negative, where a positive tag indicates permission, and a negative tag indicates denial of decryption. We resolve this tension by introducing two core technical ingredients: a *two-layer revocation mechanism* and *dynamic tag updates*.

In the *two-layer revocation mechanism*, revocation is realized via two layers of key puncturing. In the first layer, the trusted authority issues each user a secret key tied to a unique positive tag and punctured on an initial set of negative tags, enabling decryption only for ciphertexts matching the user's positive tag and negative tags distinct from those on which the key was punctured. To realize the first layer of key puncturing, we devise the KeyGen algorithm. Users may further self-revoke their access by puncturing their key on additional negative tags, as in standard PE, via the Punc algorithm. In the second layer, the trusted authority periodically publishes a time-specific period key punctured on selected positive tags, which disables decryption for those users during that period even if they retain their user keys. To realize the second layer of key puncturing, we formulate the RevAccK algorithm. We assume this time-dependent information is publicly available at the start of each period. Conceptually, we divide the power of decryption ability of the user into two key components: one key is generated for each user, and another key is generated for each time period. This key generation is carried out in a dual component-wise fashion, meaning that to decrypt or access data, a user needs both keys, provided that the user still has decryption rights after both layers of revocation.

To go beyond these revocations and to enable selective restoration and lawful delegation, we introduce *dynamic tag updates*. This allows ciphertexts to be re-tagged and/or advanced to later times without altering the underlying plaintext. These transformations are driven by time and tag-specific tokens issued by the trusted authority. We implement this mechanism by designing the Token algorithm. By updating ciphertexts using these tokens, users can regain access or get decryption rights while preserving message confidentiality. These update mechanisms are implemented by devising the UpdateCT algorithm.

Defining security for DPE, we consider the concept of indistinguishability against chosen plaintext attack (IND-CPA), and extend the selective-tag framework by having the adversary declare the challenge positive tag, negative tags, and a time period upfront. Consistent with the standard selective-tag security definition, the adversary retains the ability to learn the user keys. Furthermore, we permit the adversary to revoke positive tags of their choice, including the challenge positive tag, at any time. Notably, we permit the adversary to learn the secret key for the challenge positive tag, but only if it was revoked at the designated challenge time. As the ability to update ciphertexts and the associated concept of update tokens are not inherent to standard PE, we need to consider additional measures to ensure security guarantees. In particular, we recognize

the potential risks of misuse of update tokens, such as updating ciphertexts in the reverse direction or re-updating ciphertexts that have already been updated. To protect against these risks, a secure DPE scheme must enforce that the update token can be applied to a non-updated ciphertext with a matching positive tag, thereby transforming it into an updated ciphertext. It must not allow reverse updates or repeated updates, as the ability to update an already updated ciphertext would compromise DPE security. Together, these considerations lead to our notion of IND-sDPE-CPA, providing a unified security foundation for DPE.

Lattice-based construction of Dynamic Puncturable Encryption.
As an effort to instantiate DPE scheme, we design a lattice-based construction that is proven secure in the standard model, assuming the hardness of the Learning With Errors (LWE) problem [25]. At a high level, the design faces three intertwined challenges: (i) supporting two independent revocation mechanisms without giving any single artifact full decryption power, (ii) supporting user-side puncturing, which necessitates delegatable trapdoor structure, and (iii) enabling token-based ciphertext updates for restoration or delegation of decryption rights while preventing token misuse and related attacks. We meet these requirements through a component-wise key architecture: user keys retain a trapdoor form to enable further puncturing, yet remain insufficient on their own for decryption, while the authority publishes time-specific revocation key. This key is publicly accessible and must not possess any decryption power alone. Making these components interoperate securely is the main technical hurdle and the heart of the construction.

Our construction uses a binary tree (BT) with N leaves to manage up to N users/positive tags. Each user is assigned a leaf (positive tag) ρ, and each node θ carries an identifier $\mathbf{V}_\theta$, a user key embeds secrets for the nodes on $\mathsf{Path}(\rho)$, enabling efficient time-dependent revocation via Subset-Cover techniques. The public key include matrices $\mathbf{A}, \mathbf{D}$, while the master secret key holds trapdoors $\mathbf{T_A}, \mathbf{T_D}$. For each user tag ρ, we derive positive tag-specific matrices $\mathbf{A}_\rho, \mathbf{D}_\rho$ together with trapdoors. The user's secret key consists of two components. The first component relates to $\mathbf{A}_\rho$ and supports negative-tag puncturing through a key delegation technique from [4]. Importantly, it retains a trapdoor matrix form, allowing the user to puncture further later on. The second component pertains to $\mathbf{D}_\rho$, which binds the user to the tree path using the complete subtree method described in [20]. This involves sampling short vectors for the identifiers along $\mathsf{Path}(\rho)$ while linking $\mathbf{D}_\rho$ with $\mathbf{A}_\rho$. For the second revocation layer, the trusted authority periodically publishes a public revoke-access key for each time period t, and the revoked list of positive tags P_t, derived from a matrix $\mathbf{D}_t$ and the cover set $\mathsf{KUNodes}(\mathsf{BT}, \mathsf{P}_t)$. This update must be public and carry no standalone decryption power, yet must "plug into" the user's key to enable decryption, thereby capturing exactly the separation of power required by DPE.

Ciphertexts are encrypted under a positive tag ρ, a fixed-size set of negative tags, and a time period t, and consist of three components $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$ that share the same randomness $\mathbf{s} \in \mathbb{Z}_q^n$. The first component binds the negative tags and time, the second component is associated with a common random matrix

$\mathbf{U}$ and the message, and the third component links the positive tag with time, enabling it to interact with both revocation layers. Decryption is successful only if the user survives both revocation layers. When $\rho \notin \mathsf{P}_t$, the user combines the node in $\mathsf{Path}(\rho) \cap \mathsf{KUNodes}(\mathsf{BT}, \mathsf{P}_t)$ with the second key component to derive a transformation key. This key partially decrypts $\tilde{\mathbf{c}}$ down to an $\mathbf{A}_\rho$-bound ciphertext, which can then be fully decrypted using the first key component, provided that the user has not self-punctured the relevant negative tags.

Restoration and delegation are realized via ciphertext updates driven by the trusted authority issued tokens that are time-and-tag specific. A token can push a ciphertext from time t to $t'(> t)$ and optionally change its tags (changing the positive tag enables delegation of decryption rights; changing negative tags enables selective restoration of access). The core difficulty here is that a valid update must maintain ciphertext well-formedness while preserving the underlying plaintext and the shared randomness and without excessive noise growth. We address this with a two-part token (Δ_1, Δ_2). The first component Δ_1 updates the first part of the ciphertext to reflect the intended time and negative tags, resulting in the first part of the updated ciphertext. To generate the third part of the updated ciphertext incorporating the intended positive tag and time, we use Δ_2, which creates the third part of the updated ciphertext by taking the first and the third components of the original ciphertext as input. This process leaves the second component unchanged during the update. It is important to note that this update process maintains the common randomness. Additionally, since the randomness is represented by a uniform vector in $\mathbb{Z}_q^n$ and the encryption scheme is additively homomorphic, updated ciphertexts can be easily re-randomized.

Finally, the security reduction blends oracle-simulation ideas from lattice-based identity-based encryption [1] and key delegation technique from delegatable attribute-based encryption [4], with subtle adjustments needed to handle the two-layer revocation and token-based updates.

Organization. The rest of this paper is organized as follows. In Section 2, we present the formal definitions and security model of DPE. In Section 3, we briefly recall some background about lattices. Our lattice-based construction of DPE is described in Section 4 and analyzed in Section 5. Due to space limitations, the detailed correctness and security analysis of our lattice-based construction are deferred to the Appendix.

2 Dynamic Puncturable Encryption

We now define the syntax and security properties of DPE. As discussed in the technical overview in Section 1.1, formalizing DPE is subtle because it must reconcile two goals that naturally pull in opposite directions: fine-grained forward secrecy and controlled restoration of decryption capability. We address this challenge through two core components: a *two-layer revocation mechanism*, realized through the KeyGen and RevAccK algorithms, and *dynamic tag updates*, realized through the Token and UpdateCT algorithms. In addition, users may self-revoke access via the Punc algorithm. Together, these components capture the dynamic nature of the system. In this section, we make these ideas precise by

defining the syntax of DPE, its correctness, and introducing the security notion IND-sDPE-CPA.

Definition 1 (DPE). *A Dynamic Puncturable Encryption scheme with a message space $\mathcal{M}$, a time space $\mathcal{T}$, a positive tag space $\mathcal{P}$, and a negative tag space $\mathcal{N}$, is a tuple of algorithms* $(\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Encrypt}, \mathsf{RevAccK}, \mathsf{Decrypt}, \mathsf{Token}, \mathsf{UpdateCT})$:

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{SetUp}(1^\lambda, d)$: *On input a security parameter λ and a maximum number of negative tags d, it outputs a public key* pk *and a master secret key* msk.
- $\mathsf{sk}_{\mathsf{N}}^{\rho} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \rho, \mathsf{N})$: *On input the public key* pk, *the master secret key* msk, *a positive tag $\rho \in \mathcal{P}$ and a negative tag set* N, *it outputs a punctured secret key* $\mathsf{sk}_{\mathsf{N}}^{\rho}$.[2]
- $\mathsf{sk}_{\mathsf{N}\cup\{\eta\}}^{\rho} \leftarrow \mathsf{Punc}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}}^{\rho}, \eta)$: *On input the public key* pk, *a punctured secret key* $\mathsf{sk}_{\mathsf{N}}^{\rho}$ *and a negative tag $\eta \in \mathcal{N}$, it outputs a new punctured secret key* $\mathsf{sk}_{\mathsf{N}\cup\{\eta\}}^{\rho}$.
- $ct \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m, \rho, \boldsymbol{\eta}, t)$: *On input the public key* pk, *a message $m \in \mathcal{M}$, a positive tag $\rho \in \mathcal{P}$, a d-tuple of negative tags $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d) \in \mathcal{N}^d$, and a time period $t \in \mathcal{T}$, it outputs a ciphertext ct. For simplicity and wlog, we assume that $\rho, \boldsymbol{\eta}$ and t are efficiently computable from ct.*
- $\mathsf{rk}_{\mathsf{P}}^{t} \leftarrow \mathsf{RevAccK}(\mathsf{pk}, \mathsf{msk}, \mathsf{P}, t)$: *On input the public key* pk, *the master secret key* msk, *a time period t, a set of revoked positive tags* P *on time t, it outputs a revoke access key* $\mathsf{rk}_{\mathsf{P}}^{t}$. [3]
- $m/\perp \leftarrow \mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}}^{\rho'}, \mathsf{rk}_{\mathsf{P}}^{t}, ct, \rho, \boldsymbol{\eta}, t)$: *On input the public key* pk, *a secret key* $\mathsf{sk}_{\mathsf{N}}^{\rho'}$, *a revoke access key* $\mathsf{rk}_{\mathsf{P}}^{t}$ *on time t, and a ciphertext ct with tags $\rho, \boldsymbol{\eta}$ on time t, it outputs a plaintext m or $\perp$ if decryption fails.*
- $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2} \leftarrow \mathsf{Token}(\mathsf{pk}, \mathsf{msk}, \mathcal{E}_1, \mathcal{E}_2)$: *On input the public key* pk, *the master secret key* msk, *and two tuples $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1)$, $\mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2(> t_1))$ consisting of a positive tag, a d-tuple of negative tags and a time, it outputs a token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}$ or $\perp$.*
- $ct' \leftarrow \mathsf{UpdateCT}(\mathsf{pk}, \mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}, ct, \rho_1, \boldsymbol{\eta}_1, t_1)$: *On input the public key* pk, *a token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}$ with $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1)$, $\mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2(> t_1))$, and a ciphertext ct with tags ρ_1 and $\boldsymbol{\eta}_1 = (\eta_1, \cdots, \eta_d)$ on time t_1, it outputs the updated ciphertext ct' with tags ρ_2, $\boldsymbol{\eta}_2 = (\eta'_1, \cdots, \eta'_d)$ for time t_2 or $\perp$ indicating ct is invalid.*

Definition 2 (DPE Correctness). *A Dynamic Puncturable Encryption scheme* $(\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Encrypt}, \mathsf{RevAccK}, \mathsf{Decrypt}, \mathsf{Token}, \mathsf{UpdateCT})$ *with a message space $\mathcal{M}$, a time space $\mathcal{T}$ and a positive tag space $\mathcal{P}$, a negative tag space*

[2] Note that we sometimes use N_ρ for $\mathsf{sk}_{\mathsf{N}_\rho}^{\rho}$ to specify a set of ρ-specific punctured negative tags.

[3] Note that we sometimes use P_t for $\mathsf{rk}_{\mathsf{P}_t}^{t}$ to specify a set of t-specific punctured positive tags.

$\mathcal{N}$, *respectively, is said to be correct if for any security parameter* $\lambda \in \mathbb{N}$, *for any* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{SetUp}(1^\lambda, d)$, *for any* $\rho' \in \mathcal{P}$, *for any* $\mathsf{N}, \mathsf{N}' \subset \mathcal{N}$, *for any* $\mathsf{sk}_{\mathsf{N}}^{\rho'} \leftarrow \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \rho')$, *for any* $\eta \in \mathcal{N}$, *for any* $\mathsf{sk}_{\mathsf{N}'}^{\rho'} \leftarrow \mathsf{Punc}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}}^{\rho'}, \eta)$, *where* $\mathsf{N}' = \mathsf{N} \cup \{\eta\}$, *we require that*

1. *For any message* $m \in \mathcal{M}$, *for any* $\rho \in \mathcal{P}$, *for any* d*-tuple of negative tags* $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d) \in \mathcal{N}^d$, *for any time* $t \in \mathcal{T}$, *for any* $\mathsf{rk}_{\mathsf{P}}^t \leftarrow \mathsf{RevAccK}(\mathsf{pk}, \mathsf{msk}, \mathsf{P}, t)$, *the following holds.*
   - *If* $\rho \notin \mathsf{P} \ \wedge \ \rho' = \rho \ \wedge \ \mathsf{N}' \cap \{\eta_1, \cdots, \eta_d\} = \phi$,
   $$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}'}^{\rho'}, \mathsf{rk}_{\mathsf{P}}^t, \\ ct, \rho, \boldsymbol{\eta}, t) = m\end{array} \middle| \ ct \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m, \rho, \boldsymbol{\eta}, t)\right] = 1 - \mathsf{negl}(\lambda);$$
   - *If* $\rho \in \mathsf{P} \ \vee \ \rho' \neq \rho \ \vee \ \mathsf{N}' \cap \{\eta_1, \cdots, \eta_d\} \neq \phi$,
   $$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}'}^{\rho'}, \mathsf{rk}_{\mathsf{P}}^t, \\ ct, \rho, \boldsymbol{\eta}, t) = \perp\end{array} \middle| \ ct \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m, \rho, \boldsymbol{\eta}, t)\right] = 1 - \mathsf{negl}(\lambda);$$
   where $\mathsf{negl}(\cdot)$ *is a negligible function, and the probabilities are taken over the random coins of the algorithms.*
2. *For any message* $m \in \mathcal{M}$, *for any* $\rho_1, \rho_2 \in \mathcal{P}$, *for any* d*-tuple of negative tags* $\boldsymbol{\eta}_1 \in \mathcal{N}^d$, *and* $\boldsymbol{\eta}_2 = (\eta_1', \cdots, \eta_d') \in \mathcal{N}^d$, *for any time* $t_1, t_2 \in \mathcal{T}$, *for any* $ct \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m, \rho_1, \boldsymbol{\eta}_1, t_1)$, *for any* $\mathsf{rk}_{\mathsf{P}_{t_2}}^{t_2} \leftarrow \mathsf{RevAccK}(\mathsf{pk}, \mathsf{msk}, \mathsf{P}_{t_2}, t_2)$, *the following holds.*
   - *If* $\rho_2 \notin \mathsf{P}_{t_2} \ \wedge \ \rho' = \rho_2 \ \wedge \mathsf{N}' \cap \{\eta_1', \cdots, \eta_d'\} = \phi$,
   $$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}'}^{\rho'}, \mathsf{rk}_{\mathsf{P}_{t_2}}^{t_2}, \\ ct', \rho_2, \boldsymbol{\eta}_2, t_2) = m\end{array} \middle| \begin{array}{c}\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2} \leftarrow \mathsf{Token}(\mathsf{pk}, \mathsf{msk}, \mathcal{E}_1, \mathcal{E}_2), \\ \textit{where } \mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1), \\ \mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2(> t_1)); \\ ct' \leftarrow \mathsf{UpdateCT}(\mathsf{pk}, \mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}, ct, \rho_1, \boldsymbol{\eta}_1, t_1)\end{array}\right]$$
   $$= 1 - \mathsf{negl}(\lambda);$$
   - *If* $\rho_2 \in \mathsf{P}_{t_2} \ \vee \ \rho' \neq \rho_2 \ \vee \mathsf{N}' \cap \{\eta_1', \cdots, \eta_d'\} \neq \phi$,
   $$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}'}^{\rho'}, \mathsf{rk}_{\mathsf{P}_{t_2}}^{t_2}, \\ ct', \rho_2, \boldsymbol{\eta}_2, t_2) = \perp\end{array} \middle| \begin{array}{c}\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2} \leftarrow \mathsf{Token}(\mathsf{pk}, \mathsf{msk}, \mathcal{E}_1, \mathcal{E}_2), \\ \textit{where } \mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1), \\ \mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2(> t_1)); \\ ct' \leftarrow \mathsf{UpdateCT}(\mathsf{pk}, \mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}, ct, \rho_1, \boldsymbol{\eta}_1, t_1)\end{array}\right]$$
   $$= 1 - \mathsf{negl}(\lambda);$$
   where $\mathsf{negl}(\cdot)$ *is a negligible function, and the probabilities are taken over the random coins of the algorithms.*

Security Game of DPE for Selective Tags against Chosen Plaintext Attack (IND-sDPE-CPA): Let $\mathcal{A}$ be the PPT adversary and $\Pi = (\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Punc}, \mathsf{Encrypt}, \mathsf{RevAccK}, \mathsf{Decrypt}, \mathsf{Token}, \mathsf{UpdateCT})$ be a DPE scheme with a message space $\mathcal{M}$, time space $\mathcal{T}$, a positive tag space $\mathcal{P}$ and a negative tag space $\mathcal{N}$. Security game is defined according to the following game $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{IND\text{-}sDPE\text{-}CPA}}(1^\lambda)$:

1. **Initial**: $\mathcal{A}$ sends the target positive tag $\rho^* \in \mathcal{P}$, the target d-tuple of negative tags $\boldsymbol{\eta}^* = (\eta_1^*, \cdots, \eta_d^*) \in \mathcal{N}^d$, and time $t^* \in \mathcal{T}$ to the challenger. Let P^* be the set of revoked positive tags at time t^*.

2. **Set Up**: The challenger runs $\mathsf{SetUp}(1^\lambda, d)$ to get $(\mathsf{pk}, \mathsf{msk})$ and give pk to $\mathcal{A}$. It initializes three empty sets $P, C, \overline{C}$ to record puncture key queries, corrupt key queries, and token queries. Also, the challenger introduces a counter numCT to 0, a key-value store $\mathcal{H}$ to be empty, and a set Derive to be empty.

3. **Query Phase 1**: The adversary $\mathcal{A}$ may make following queries polynomially many times:

- $\mathcal{Q}_{Puncture}(\rho, \eta)$: Given a positive tag $\rho \in \mathcal{P}$, a negative tag $\eta \in \mathcal{N}$, the challenger updated the negative tag set N_ρ, on which the secret key has been punctured, as $\mathsf{N}_\rho := \mathsf{N}_\rho \cup \{\eta\}$, and computes $\mathsf{sk}^\rho_{\mathsf{N}_\rho}$ by running $\mathsf{KeyGen}, \mathsf{Punc}$ algorithms accordingly. Update the tuple $(\rho, \mathsf{sk}^\rho_{\mathsf{N}_\rho}, \mathsf{N}_\rho)$ from the set P.
- $\mathcal{Q}_{Corrupt}(\rho)$: The challenger consider the following cases.

  Case-1: $\rho \neq \rho^*$
  - The first time the adversary issues this query for ρ, the challenger do as follows:
    - It returns $\perp$, if there was a query to $\mathcal{Q}_{RevAccK}(t, \mathsf{P_t})$ with $\rho \notin \mathsf{P}_t$ and a token query $\mathcal{Q}_{Token}(\rho^*, \boldsymbol{\eta}^*, t^* \to \rho, \boldsymbol{\eta}, t)$, where $\boldsymbol{\eta} \cap \mathsf{N}_\rho = \phi$. Add $(\rho, \perp, \mathsf{N}_\rho)$ to the set C.
    - Otherwise, it returns the most recent secret key $\mathsf{sk}^\rho_{\mathsf{N}_\rho}$. Add $(\rho, \mathsf{sk}^\rho_{\mathsf{N}_\rho}, \mathsf{N}_\rho)$ to the set C.
  - All subsequent queries for this positive tag ρ return $\perp$.

  Case-2: $\rho = \rho^*$
  - The first time the adversary issues this query for ρ^*, the challenger do as follows:
    - It returns $\perp$, if any of the following holds and add $(\rho^*, \perp, \mathsf{N}_{\rho^*})$ to C.
      1. $\mathsf{N}_{\rho^*} \cap \{\eta_1^*, \cdots, \eta_d^*\} = \emptyset$ and there was a query to $\mathcal{Q}_{RevAccK}(t^*, \mathsf{P}_{t^*})$ with $\rho^* \notin \mathsf{P}_{t^*}$.
      2. If there was a query to $\mathcal{Q}_{RevAccK}(t, \mathsf{P}_t)$ with $\rho^* \notin \mathsf{P}_t$ and a token query $\mathcal{Q}_{Token}(\rho^*, \boldsymbol{\eta}^*, t^* \to \rho^*, \boldsymbol{\eta}, t)$, where $\boldsymbol{\eta} \cap \mathsf{N}_{\rho^*} = \phi$.
    - Otherwise, it returns the most recent secret key $\mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}$ to the adversary. Add $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}, \mathsf{N}_{\rho^*})$ to the set C.
  - All subsequent queries for this positive tag ρ^* return $\perp$.
- $\mathcal{Q}_{RevAccK}(t, \mathsf{P_t})$: On input time t, a list of revoked positive tags P_t, the challenger outputs the revoke access key $\mathsf{rk}^t_{\mathsf{P}_t}$ for time t satisfying the following.
  - If there was a query to $\mathcal{Q}_{Corrupt}(\rho^*)$ with $\mathsf{N}_{\rho^*} \cap \{\eta_1^*, \cdots, \eta_d^*\} = \emptyset$, then ρ^* must be in P_{t^*}.
  - If there was a query to $\mathcal{Q}_{Token}(\rho^*, \boldsymbol{\eta}^*, t^* \to \rho, \boldsymbol{\eta}, t(> t^*))$ and a corrupt query on ρ with $\mathsf{N}_\rho \cap \boldsymbol{\eta} = \phi$, then ρ must be in P_t.

  Note that the adversary is allowed to query only one revoked access key at each time t. If $t = t^*$, then P^* must be a subset of the P_{t^*} at t^*.
- $\mathcal{Q}_{Token}(\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1) \to \mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2))$: Given two positive tags $\rho_1, \rho_2 \in \mathcal{P}$, and two d-tuple of negative tags $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \mathcal{N}^d$, and time periods t_1, $t_2(> t_1)$, the challenger consider the following two cases:
  - For $(\rho_1, \boldsymbol{\eta}_1, t_1) \neq (\rho^*, \boldsymbol{\eta}^*, t^*)$, it returns a token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}$ and add it to $\overline{C}$.

- For $(\rho_1, \boldsymbol{\eta}_1, t_1) = (\rho^*, \boldsymbol{\eta}^*, t^*)$, it returns a token if there was no query to $\mathcal{Q}_{Corrupt}(\rho_2)$ with $\boldsymbol{\eta}_2 \cap \mathsf{N}_{\rho_2} = \phi$ and no $\mathcal{Q}_{RevAccK}(t_2, \mathsf{P}_{t_2})$ with $\rho_2 \notin \mathsf{P}_{t_2}$. Add token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}$ to $\overline{C}$. Otherwise, it returns $\perp$.

– $\mathcal{Q}_{\mathsf{Enc}}(\rho, \boldsymbol{\eta}, m, t)$: Given a positive tag ρ, a d-tuple of negative tags $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)$, and message m, time period t, outputs ciphertext ct by running $\mathsf{Encrypt}(\cdot)$. Increment $numCt$ and add ct to the set $\mathcal{H}$ with key $(\rho, \boldsymbol{\eta}, t, numCt)$.
– $\mathcal{Q}_{\mathsf{Update}}$: Given two positive tags ρ_1, ρ_2, two d-tuple of negative tags $\boldsymbol{\eta}_1 = \{\eta_1, \cdots, \eta_d\}, \boldsymbol{\eta}_2 = \{\eta'_1, \cdots, \eta'_d\}$ and key $(\rho_1, \boldsymbol{\eta}_1, t, k)$, where $k \leq numCt$, outputs $\perp$ if there is no value in $\mathcal{H}$ with key $(\rho_1, \boldsymbol{\eta}_1, t, k)$. Otherwise, let ct be that value in $\mathcal{H}$ and outputs updated ciphertext ct' by running $\mathsf{UpdateCT}(\cdot)$.

4. **Challenge**: $\mathcal{A}$ submits two messages $m_0, m_1 \in \mathcal{M}$ to the challenger. The challenger outputs a challenge ciphertext $ct_\beta \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m_\beta, \rho^*, \boldsymbol{\eta}^*, t^*)$ for either $\beta = 0$ or $\beta = 1$, by choosing a random bit $\beta \in \{0, 1\}$. Increment $numCt$ and add $numCt$ to the set $Derive$. Store the value ct^*_β to the set $\mathcal{H}$ with key $(\rho^*, \boldsymbol{\eta}^*, t^*, numCt)$.

5. **Query Phase 2**: This phase is identical to **Query Phase 1**. After receiving the challenge ciphertext, $\mathcal{A}$ continues to have access to the $\mathcal{Q}_{\mathsf{Puncture}}, \mathcal{Q}_{\mathsf{Corrupt}}$, $\mathcal{Q}_{\mathsf{RevAccK}}$, $\mathcal{Q}_{\mathsf{Token}}$, $\mathcal{Q}_{\mathsf{Enc}}$, and $\mathcal{Q}_{\mathsf{Update}}$ as in **Query Phase 1** except the following constraints for $\mathcal{Q}_{\mathsf{Update}}$ oracle: Outputs $\perp$ if $k \in Derive$.

6. **Guess**: On input β' from $\mathcal{A}$, this oracle outputs 1 if $\beta = \beta'$ and 0 otherwise. The advantage of an adversary in the above experiment $\mathsf{Exp}^{\mathsf{IND\text{-}sDPE\text{-}CPA}}_{\mathcal{A}}(1^\lambda)$ is defined as $|\mathsf{Pr}[\beta' = \beta] - \frac{1}{2}|$.

Definition 3. *A* DPE *scheme is* IND-sDPE-CPA *secure if all* PPT *adversaries* $\mathcal{A}$ *have at most a negligible advantage in experiment* $\mathsf{Exp}^{\mathsf{IND\text{-}sDPE\text{-}CPA}}_{\mathcal{A}}(1^\lambda)$.

Remark 1. To prevent the adversary $\mathcal{A}$ from trivially winning the security game, the above game has some restrictions on the corrupt queries, the revoke access key queries, and the token queries, respectively. Here, we discuss the rationale behind the conditions adopted in the security model above. Intuitively, for a tuple $(m_0, m_1, \rho^*, \boldsymbol{\eta}^*, t^*)$ such that $m_0 \neq m_1$, issued by $\mathcal{A}$ at challenge phase to obtain the ct_β, and $\mathcal{A}$ issues some specific queries to $\mathcal{Q}_{Corrupt}, \mathcal{Q}_{RevAccK}, \mathcal{Q}_{Token}$. Then, $\mathcal{A}$ could trivially know the challenge bit β in the following two cases.
Case-1: $\mathcal{A}$ issues the query $\mathcal{Q}_{Corrupt}(\rho^*)$ and get $\mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}$ such that $\mathsf{N}_{\rho^*} \cap \boldsymbol{\eta}^* = \phi$, and the query $\mathcal{Q}_{RevAccK}(t^*, \mathsf{P}_{t^*})$, where $\rho^* \notin \mathsf{P}_{t^*}$.
Case-2: $\mathcal{A}$ issues a token query $\mathcal{Q}_{Token}(\rho^*, \boldsymbol{\eta}^*, t^* \to \rho, \boldsymbol{\eta}, t)$ and a query $\mathcal{Q}_{Corrupt}(\rho)$ and get $\mathsf{sk}^{\rho}_{\mathsf{N}_\rho}$ such that $\mathsf{N}_\rho \cap \boldsymbol{\eta} = \phi$ and the query $\mathcal{Q}_{RevAccK}(t, \mathsf{P}_t)$, where $\rho \notin \mathsf{P}_t$.

In case-1, since $\rho^* \notin \mathsf{P}_{t^*}$ and $\mathsf{N}_{\rho^*} \cap \boldsymbol{\eta}^* = \phi$, with the revoke access key for time t^* and the secret key of ρ^*, $\mathcal{A}$ can correctly decrypt the message m_β and get to know about the challenge bit β.

In case-2, since $\mathcal{A}$ has a token $\mathsf{tk} \leftarrow \mathcal{Q}_{Token}(\rho^*, \boldsymbol{\eta}^*, t^* \to \rho, \boldsymbol{\eta}, t)$, with this token tk, $\mathcal{A}$ can update the challenge ciphertext ct_β to ct'_β. Note that ct'_β is with tags $\rho, \boldsymbol{\eta}$ and time t. Since, $\mathcal{A}$ has the revoke access key for time t and the secret key of ρ satisfying $\rho \notin \mathsf{P}_t$ and $\mathsf{N}_\rho \cap \boldsymbol{\eta} = \phi$, $\mathcal{A}$ can correctly decrypt ct'_β with the

revoke access key for time t and the secret key of ρ, and get to know about the message m_β, so about the challenge bit β.

In order to avoid these trivial wins, we enforced the restrictions in $\mathcal{Q}_{Corrupt}$, $\mathcal{Q}_{RevAccK}$, and $\mathcal{Q}_{Token}$.

3 Prerequisites for the construction

We denote the real numbers and the integers by $\mathbb{R}, \mathbb{Z}$, respectively. We denote column-vectors by lower-case bold letters (e.g. $\mathbf{b}$), so row-vectors are represented via transposition (e.g. $\mathbf{b}^t$). Matrices are denoted by upper-case bold letters and treat a matrix $\mathbf{X}$ interchangeably with its ordered set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ of column vectors. We use $\mathbf{I}$ for the identity matrix and $\mathbf{0}$ for the zero matrix, where the dimension will be clear from context. We use $[*|*]$ to denote the concatenation of vectors or matrices. The *statistical distance* between two distributions $\mathbf{X}$ and $\mathbf{Y}$ over a countable domain Ω defined as $\frac{1}{2}\sum_{w\in\Omega}|\mathsf{Pr}[\mathbf{X}=w]-\mathsf{Pr}[\mathbf{Y}=w]|$. We say that a distribution over Ω is ϵ-far if its statistical distance from the uniform distribution is at most ϵ.

Lattices: A *lattice* Λ is a discrete additive subgroup of $\mathbb{R}^m$. Specially, a lattice Λ in $\mathbb{R}^m$ with basis $\mathbf{B} = \left[\mathbf{b}_1 | \cdots | \mathbf{b}_n\right] \in \mathbb{R}^{m\times n}$ is defined as $\Lambda := \{\sum_{i=1}^{n} \mathbf{b}_i x_i | x_i \in \mathbb{Z}\ \forall i = 1, \ldots, n\} \subseteq \mathbb{R}^m$. We call n the rank of Λ, and if $n = m$, we say that Λ is a full rank lattice. In this paper, we mainly consider full rank lattices containing $q\mathbb{Z}^m$, called q-ary lattices, defined as: for a given matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, $\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \bmod q\}$; $\Lambda_q(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \exists\ \mathbf{s} \in \mathbb{Z}_q^n\ s.t.\ \mathbf{z} = \mathbf{A}^\top \mathbf{s} \bmod q\}$; $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{u} \bmod q\} = \Lambda_q^\perp(\mathbf{A}) + \mathbf{x}\ for\ \mathbf{x} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})\}$.

Matrix Norms: For a vector $\mathbf{u}$, we let $\|\mathbf{u}\|$ denotes its ℓ_2 norm. For a matrix $\mathbf{R} \in \mathbb{Z}^{k\times m}$, let $\widetilde{\mathbf{R}}$ be the result of applying Gram-Schmidt (GS) orthogonalization to the columns of $\mathbf{R}$. We denote three matrix norms as: $\|\mathbf{R}\|$ denotes the ℓ_2 length of the longest column of $\mathbf{R}$; $\|\mathbf{R}\|_{\mathrm{GS}} = \left\|\widetilde{\mathbf{R}}\right\|$, where $\widetilde{\mathbf{R}}$ is the GS orthogonalization of $\mathbf{R}$; $\|\mathbf{R}\|_2$ is the operator norm of $\mathbf{R}$ defined as $\|\mathbf{R}\|_2 = \sup_{\|\mathbf{x}\|=1}\|\mathbf{Rx}\|$.

Gaussian on Lattices: Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. For a vector $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter $\sigma \in \mathbb{R}$, define: $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp\left(\pi\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$ and $\rho_{\mathbf{c},\sigma}(\Lambda) = \sum_{\mathbf{x}\in\mathbf{\Lambda}}\rho_{\mathbf{c},\sigma}(\mathbf{x})$. The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ is $\mathcal{D}_{\mathbf{c},\sigma}(\Lambda)(\mathbf{y}) = \frac{\rho_{\mathbf{c},\sigma}(\mathbf{y})}{\rho_{\mathbf{c},\sigma}(\Lambda)}, \forall \mathbf{y} \in \Lambda$.

Lemma 1 ([4], Lemma 2.5**).** *Let $n, m, k, q, \sigma > 0$ and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{U} \in \mathbb{Z}_q^{n\times k}$. If $\mathbf{R} \in \mathbb{Z}^{m\times k}$ is sampled from $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$ and $\mathbf{S}$ is sampled uniformly in $\{+1, -1\}^{m\times m}$, the followings hold with overwhelming probability in m:*

$$\left\|\mathbf{R}^\top\right\|_2 \le \sigma\sqrt{mk}, \quad \|\mathbf{R}\|_2 \le \sigma\sqrt{mk} \quad and \quad \|\mathbf{S}\|_2 \le 20\sqrt{m}.$$

Learning With Errors (LWE) [24]: The Learning with Errors (LWE) problem was introduced by Regev [24]. Here, we define the decisional version of LWE. The security of our scheme is based on this hardness assumption.

Definition 4 (Decisional LWE (dLWE)). *Consider a prime integer q, positive integers n, m, and a noise distribution χ over $\mathbb{Z}_q$. The $\mathsf{dLWE}_{n,m,q,\chi}$ problem is*

to distinguish the two distributions $(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e})$ *and* $(\mathbf{A}, \mathbf{u})$*, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$*,* $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$*,* $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ *and* $\mathbf{e} \xleftarrow{\$} \chi^m$ *are sampled.*

Let the noise distribution χ is B- bounded if its support is in $[-B, B]$. For any constant $d > 0$ and sufficiently large q, Regev [24] through a quantum reduction showed that taking χ as a q/n^d-bounded discretized Gaussian distribution, the $\mathsf{dLWE}_{n,m,q,\chi}$ problem is as hard as approximating the worst-case $GapSVP$ to $n^{O(d)}$ factors, which is believed to be hard. In subsequent works, (partial) dequantization of Regev's reduction was achieved [5,21]. More generally, let $\chi_{max} < q$ be the bound on the noise distribution. The difficulty of the problem is measured by the ratio q/χ_{max}. The problem appears to remain hard even when $q/\chi_{max} < 2n^\epsilon$ for some fixed ϵ that is $0 < \epsilon < 1/2$. We refer the reader to [24,21,4,6] for more information.

Trapdoor Generators and Related Algorithms: Here, we briefly describe the properties of algorithms for generating a short basis of lattices and algorithms for finding a low-norm matrix $\mathbf{X} \in \mathbb{Z}^{m \times k}$ such that $\mathbf{AX} = \mathbf{U}$.

Lemma 2. *Let* $n, m, q > 0$ *be integers with* q *prime. There are polynomial time algorithms as follows:*

1. $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ *[2,3,19]: A randomized algorithm that, when* $m = \Theta(n \log q)$*, outputs a full-rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, and a basis* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ *for* $\Lambda_q^\perp(\mathbf{A})$ *such that* $\mathbf{A}$ *is* $\mathsf{negl}(\lambda)$*-close to uniform and* $\|\mathbf{T}\|_{GS} = O(\sqrt{n \log q})$ *with all but negligible probability in* n*.*
2. $\mathbf{T}_{(\mathbf{A}|\mathbf{B})} \leftarrow \mathsf{ExtendRight}(\mathbf{A}, \mathbf{T_A}, \mathbf{B})$ *[7]: A deterministic algorithm that given full-rank matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$*, and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$ *outputs a basis* $\mathbf{T}_{(\mathbf{A}|\mathbf{B})}$ *of* $\Lambda_q^\perp(\mathbf{A}|\mathbf{B})$ *such that* $\|\mathbf{T_A}\|_{GS} = \|\mathbf{T}_{(\mathbf{A}|\mathbf{B})}\|_{GS}$*.*
3. $\mathbf{T_M} \leftarrow \mathsf{ExtendLeft}(\mathbf{A}, \mathbf{G}, \mathbf{T_G}, \mathbf{R})$*, where* $\mathbf{M} = \left[\mathbf{A} \middle| \mathbf{G} + \mathbf{AR}\right]$ *[1]: A deterministic algorithm that given full-rank matrices* $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n \times m}$*, and a basis* $\mathbf{T_G}$ *of* $\Lambda_q^\perp(\mathbf{G})$ *outputs a basis* $\mathbf{T_M}$ *of* $\Lambda_q^\perp(\mathbf{M})$ *such that* $\|\mathbf{T_M}\|_{GS} \leq \|\mathbf{T_G}\|_{GS} \cdot (1 + \|\mathbf{R}\|_2)$*.*

Lemma 3 ([4], Lemma 2.6**).** *Let* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ *be a basis for* $\Lambda_q^\perp(\mathbf{A})$ *and* $\mathbf{U} \in \mathbb{Z}_q^{n \times k}$*. There are polynomial time algorithms that output* $\mathbf{X} \in \mathbb{Z}^{m \times k}$ *satisfying* $\mathbf{AX} = \mathbf{U}$ *with the properties below:*

1. $\mathbf{X} \leftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{T_A}, \mathbf{U}, \sigma)$ *[15]: A randomized algorithm that, when* $\sigma = \|\mathbf{T_A}\|_{GS} \cdot \omega(\sqrt{\log m})$*, outputs a random sample* $\mathbf{X}$ *from a distribution that is statistically close to* $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}(\mathbf{A}))$*.*
2. $\mathbf{T}_{\mathbf{A}}' \leftarrow \mathsf{RandBasis}(\mathbf{A}, \mathbf{T_A}, \sigma)$ *[7]: A randomized algorithm that, when* $\sigma = \|\mathbf{T_A}\|_{GS} \cdot \omega(\sqrt{\log m})$*, outputs a basis* $\mathbf{T}_{\mathbf{A}}'$ *of* $\Lambda_q^\perp(\mathbf{A})$ *sampled from a distribution that is statistically close to* $(\mathcal{D}_\sigma(\Lambda_q^\perp(\mathbf{A})))^m$*. Here* $\|\mathbf{T}_{\mathbf{A}}'\|_{GS} < \sigma\sqrt{m}$ *with all but negligible probability.*

Lemma 4 ([4], Lemma 2.8**).**

1. $\mathbf{X} \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{T_A}, \mathbf{B}, \mathbf{U}, \sigma)$*: A randomized algorithm that given full-rank matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$*, and a matrix* $\mathbf{U} \in \mathbb{Z}_q^{n \times m}$*, a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *and* $\sigma = \|\mathbf{T_A}\|_{GS} \cdot \omega(\sqrt{\log m})$*, outputs a random sample* $\mathbf{X} \in \mathbb{Z}^{2m \times m}$ *from a distribution that is statistically close to* $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}((\mathbf{A}|\mathbf{B})))$*. This algorithm is the composition of two algorithms:* $\mathbf{T}_{(\mathbf{A}|\mathbf{B})} \leftarrow \mathsf{ExtendRight}(\mathbf{A}, \mathbf{T_A}, \mathbf{B})$ *and* $\mathbf{X} \leftarrow \mathsf{SampleD}((\mathbf{A}|\mathbf{B}), \mathbf{T}_{(\mathbf{A}|\mathbf{B})}, \mathbf{U}, \sigma)$.
2. $\mathbf{X} \leftarrow \mathsf{SampleLeft}(\mathbf{A}, \mathbf{S}, y, \mathbf{U}, \sigma)$*: A randomized algorithm that given full-rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, and matrices* $\mathbf{S}, \mathbf{U} \in \mathbb{Z}_q^{n \times m}$*,* $y \neq 0 \in \mathbb{Z}_q$ *and* $\sigma = \sqrt{5} \cdot (1 + \|\mathbf{S}\|_2) \cdot \omega(\sqrt{\log m})$*, outputs a random sample* $\mathbf{X} \in \mathbb{Z}^{2m \times m}$ *from a distribution that is statistically close to* $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{U}}((\mathbf{A}|y\mathbf{G} + \mathbf{AS})))$*. This algorithm is the composition of two algorithms:* $\mathbf{T}_{(\mathbf{A}|y\mathbf{G}+\mathbf{AS})} \leftarrow \mathsf{ExtendLeft}(\mathbf{A}, y\mathbf{G}, \mathbf{T_G}, \mathbf{S})$ *and* $\mathbf{X} \leftarrow \mathsf{SampleD}((\mathbf{A}|y\mathbf{G} + \mathbf{AS}), \mathbf{T}_{(\mathbf{A}|y\mathbf{G}+\mathbf{AS})}, \mathbf{U}, \sigma)$.

Next, we define three types of evaluation algorithms from [4]. Let n and $q = q(n)$, and $m = \Theta(n \log q)$ be positive integers. Let $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ be the fixed matrix. For $x \in \mathbb{Z}_q, \mathbf{B} \in \mathbb{Z}_q^{n \times m}, \mathbf{s} \in \mathbb{Z}_q^n$, and $\delta > 0$ define the set $E_{\mathbf{s},\delta}(x, \mathbf{B}) = \{(x\mathbf{G} + \mathbf{B})^\top \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m, \text{where} \|\mathbf{e}\| < \delta\}$.

Lemma 5 (Evaluation Algorithms [4]). *The three efficient deterministic evaluation algorithms* $\mathsf{Eval}_{\mathsf{pk}}, \mathsf{Eval}_{ct}, \mathsf{Eval}_{sim}$ *satisfy the following properties with respect to the family of functions* $\mathcal{F} = \{f : (\mathbb{Z}_q)^\ell \to \mathbb{Z}_q\}$*, in which each function can be computed by some circuit of a family of depth* d*, polynomial-size arithmetic circuits* $(\mathcal{C}_\lambda)_{\lambda \in \mathbb{N}}$ *and a positive integer-valued function* $\alpha_{\mathcal{F}} : \mathbb{Z} \to \mathbb{Z}$*:*

1. $\mathbf{B}_f \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f \in \mathcal{F}, \{\mathbf{B}_i\}_{i=1}^\ell)$*, where* $\mathbf{B}_f$ *and each* $\mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$.
2. $\mathbf{c}_f \leftarrow \mathsf{Eval}_{ct}(f \in \mathcal{F}, \{x_i, \mathbf{B}_i, \mathbf{c}_i\}_{i=1}^\ell)$*, where* $\mathbf{c}_f \in \mathbb{Z}_q^m$*, and each* $x_i \in \mathbb{Z}_q, \mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$*, and* $\mathbf{c}_i \in E_{\mathbf{s},\delta}(x_i, \mathbf{B}_i)$ *for some* $\mathbf{s} \in \mathbb{Z}_q^n$ *and* $\delta > 0$*. The output* $\mathbf{c}_f$ *must satisfy* $\mathbf{c}_f \in E_{\mathbf{s},\Delta}(f(\mathbf{x}), \mathbf{B}_f)$*, where* $\mathbf{B}_f \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f \in \mathcal{F}, \{\mathbf{B}_i\}_{i=1}^\ell)$*,* $\mathbf{x} = (x_1, \cdots, x_\ell)$*, and* $\Delta < \delta \cdot \alpha_{\mathcal{F}}(n)$*, where* $\alpha_{\mathcal{F}}(n)$ *measures the increase in the noise magnitude in* $\mathbf{c}_f$ *compared to the input ciphertext.*
3. $\mathbf{R}_f \leftarrow \mathsf{Eval}_{sim}(f \in \mathcal{F}, \{x_i^*, \mathbf{R}_i\}_{i=1}^\ell, \mathbf{A})$*, where* $\mathbf{R}_f$ *and each* $\mathbf{R}_i \in \mathbb{Z}_q^{m \times m}$*, each* $x_i^* \in \mathbb{Z}_q$*. For* $\mathbf{x}^* = (x_1^*, \cdots, x_\ell^*)$*,* $\mathbf{R}_f$ *satisfies* $\mathbf{AR}_f - f(\mathbf{x}^*)\mathbf{G} = \mathbf{B}_f$*, where* $\mathbf{B}_f \leftarrow \mathsf{Eval}_{\mathsf{pk}}(f \in \mathcal{F}, \{\mathbf{AR}_i - x_i^*\mathbf{G}\}_{i=1}^\ell)$*. For all* $f \in \mathcal{F}$*, and for* $\mathbf{R}_1, \cdots, \mathbf{R}_\ell \xleftarrow{\$} \{+1, -1\}^{m \times m}$*,* $\|\mathbf{R}_f\|_2 < \alpha_{\mathcal{F}}(n)$ *with all but negligible probability.*

The Complete Subtree Method: The complete subtree (CS) method, proposed by Naor, Naor, and Lotspiech [20], is commonly used in revocation systems. This method utilizes a node selection algorithm known as KUNodes. In this algorithm, we construct a complete binary tree (BT) with at least N leaf nodes, where N represents the maximum number of positive tags in the system. Each positive tag corresponds to a leaf node in the binary tree. We use the following notation: if θ is a non-leaf node, θ_l and θ_r represent the left and right children of θ, respectively. Whenever ρ is a leaf node, the set $\mathsf{Path}(\rho)$ denotes the collection of nodes from θ to the root, including both θ and the root node. The KUNodes algorithm takes two inputs: the binary tree BT and a revocation list P_t, which contains the revoked positive tags at time t. The algorithm outputs a set of nodes $\mathbf{Y}$, which is the smallest subset of nodes containing an ancestor

for all the leaf nodes corresponding to the non-revoked active positive tags. It is known that the size of the set $\mathbf{Y}$, generated by $\mathsf{KUNodes}(\mathsf{BT}, \mathsf{P}_t)$, is at most $r \log \frac{N}{r}$, where r is the number of positive tags in P_t. A detailed description of the $\mathsf{KUNodes}$ algorithm is provided below.

$\mathsf{KUNodes}(\mathsf{BT}, \mathsf{P}_t)$

$\mathbf{X}, \mathbf{Y} \leftarrow \emptyset$

$\forall \rho_i \in \mathsf{P}_t$: add $\mathsf{Path}(\rho_i)$ to $\mathbf{X}$

$\forall \theta \in \mathbf{X}$: if $\theta_l \notin \mathbf{X}$, then add θ_l to $\mathbf{Y}$;
if $\theta_r \notin \mathbf{X}$, then add θ_r to $\mathbf{Y}$

If $\mathbf{Y} = \emptyset$, then add root to $\mathbf{Y}$

Return $\mathbf{Y}$.

4 The proposed construction of Dynamic Puncturable Encryption (DPE)

In this section, we present our lattice-based construction of DPE. As discussed in the technical overview in Section 1.1, the main challenge in instantiating DPE is to support two independent revocation mechanisms and user-side puncturing without granting any single artifact full decryption power, while also enabling token-based ciphertext updates without misuse. At a high level, KeyGen outputs a two-component user secret key: the first component is a trapdoor-based key that supports user-side negative-tag puncturing via Punc, while the second binds the user's positive tag to a binary-tree path, enabling efficient time-dependent access control. $\mathsf{RevAccK}$ periodically publishes a public revoke-access key, derived from $\mathsf{KUNodes}(\cdot)$. This public information has no standalone decryption power, but is required by Dec, so that decryption succeeds only when both revocation layers permit it. To support controlled restoration and delegation of decryption rights via ciphertext updates, Token issues time and tag specific update tokens, and $\mathsf{UpdateCT}$ uses these tokens to re-tag ciphertexts or/and advance them to later times. The main challenge is to perform these updates while preserving ciphertext well-formedness, the underlying plaintext, and the shared randomness, without incurring excessive noise growth. We address this using a two-part token (Δ_1, Δ_2) that updates the ciphertext component-wise while preserving the plaintext and controlling noise growth.

We begin the construction by setting the parameters as follows.

- $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is a gadget matrix for integer n, large enough prime power $q = poly(n)$, $m = \Theta(n \log q)$ and $k = \lceil \log q \rceil$.
- Let $d(< q)$ be the maximum number of tags per ciphertext.
- Consider the message space is $\mathcal{M} = \{0,1\}^m$, the time space is $\mathcal{T} \subset \mathbb{Z}_q^n$ and the positive tag space is $\mathcal{P} = \mathbb{Z}_q^n$, the negative tag space is $\mathcal{N} = \mathbb{Z}_q$.
- Choose $N = poly(\lambda)$ as the maximum number of positive tags that the system will support. Obtain a binary tree BT with at least N leaf nodes.

– Full-Rank Difference (FRD) map [1]: In the following construction, we use an encoding function with *full-rank difference*, defined as FRD: $\mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$, mapping $\rho \mapsto \mathbf{H}_\rho$. FRD satisfies the following properties:
  1. $\forall$ *distinct* $\rho_1, \rho_2 \in \mathbb{Z}_q^n$, the matrix $\mathbf{H}_{\rho_1} - \mathbf{H}_{\rho_2} \in \mathbb{Z}_q^{n\times n}$ is full rank;
  2. $\forall\ \rho \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$, the matrix $\mathbf{H}_\rho \in \mathbb{Z}_q^{n\times n}$ is full rank;
  3. FRD is computable in polynomial time (in $n \log q$).
– Let χ be a χ_{max}-bounded distribution for which $\mathsf{dLWE}_{n,3m,q,\chi}$ is hard.
– Initially, set $\sigma_0 = \omega(\alpha_{\mathcal{F}} \cdot \sqrt{\log m})$.
– Let $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)$. We define the family of functions $\mathcal{F} = \{f_\eta \mid f_\eta : \mathbb{Z}_q^d \to \mathbb{Z}_q, \forall \eta \in \mathbb{Z}_q\}$, where $f_\eta(\boldsymbol{\eta}) \neq 0 \mod q$ if $\eta \in \{\eta_1, \cdots, \eta_d\}$, otherwise $f_\eta(\boldsymbol{\eta}) = 0 \mod q$.

The proposed DPE consists of the following algorithms:

$\mathsf{SetUp}(1^\lambda, d)$: On input a security parameter λ, the maximum number of tags d with each ciphertext, do as follows:

1. Generate two independent pairs $(\mathbf{A}, \mathbf{T_A}), (\mathbf{D}, \mathbf{T_D})$ by using $\mathsf{TrapGen}(1^n, 1^m, q)$, where $\mathbf{A}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n\times m}$, and $\mathbf{T_A}, \mathbf{T_D} \in \mathbb{Z}_q^{m\times m}$, bases of $\Lambda_q^\perp(\mathbf{A})$, $\Lambda_q^\perp(\mathbf{D})$, respectively.
2. Choose $d+5$ uniformly random matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d, \mathbf{D}_0, \mathbf{D}_1, \mathbf{U} \in \mathbb{Z}_q^{n\times m}$.
3. Outputs a public key $\mathsf{pk} = \{\mathbf{A}, \mathbf{D}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d, \mathbf{D}_0, \mathbf{D}_1, \mathbf{U}, \mathbf{G}\}$, a master secret key $\mathsf{msk} = \{\mathbf{T_A}, \mathbf{T_D}\}$.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \rho, \mathsf{N})$: On input the public key pk, the master secret key msk, a positive tag $\rho \in \mathcal{P}$ and a negative tag set $\mathsf{N} = \{\eta_1, \cdots, \eta_\ell\}$, where each $\eta_i \in \mathbb{Z}_q$, and a positive tag $\rho \in \mathbb{Z}_q^n$, do as follows:

1. Let $\mathbf{H}_\rho$ be the encoding of ρ. Set $\mathbf{A}_\rho = \mathbf{A}_0 + \mathbf{H}_\rho \mathbf{G}$, $\mathbf{D}_\rho = \mathbf{D}_0 + \mathbf{H}_\rho \mathbf{G}$.
2. For a negative tag set $\mathsf{N} = \{\eta_1, \cdots, \eta_\ell\}$, evaluate $\mathbf{B}_{f_\eta} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\{\mathbf{B}_i\}_{i=1}^d, f_\eta)$ for each $\eta_i \in \mathbb{Z}_q$. Compute $\mathbf{T}_{\mathsf{N}}^{\rho,\mathrm{ER}} \leftarrow \mathsf{ExtendRight}\left(\mathbf{A}, \mathbf{T_A}, \left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_\ell}}\right]\right)$. Compute the re-randomized trapdoor $\mathbf{T}_{\mathsf{N}}^\rho$ for $\left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_\ell}}\right]$ as $\mathbf{T}_{\mathsf{N}}^\rho \leftarrow \mathsf{RandBasis}\left(\left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_\ell}}\right], \mathbf{T}_{\mathsf{N}}^{\rho,\mathrm{ER}}, \sigma_{\ell+1}\right)$, where $\sigma_{\ell+1} = \sigma_0 \cdot (\sqrt{m \log m})^\ell$. Set $\mathsf{sk}_{1,\mathsf{N}}^\rho = \mathbf{T}_{\mathsf{N}}^\rho$.
3. $\forall\ \theta \in Path(\rho)$, if $\mathbf{V}_\theta$ is undefined, then pick $\mathbf{V}_\theta \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and store it on θ. Sample $\mathbf{E}_{1,\theta} \leftarrow \mathsf{SampleRight}\left(\mathbf{D}, \mathbf{D}_\rho, \mathbf{T_D}, \mathbf{A}_\rho - \mathbf{V}_\theta, \sigma_0\right)$ for each $\theta \in Path(\rho)$. Note that $\mathbf{E}_{1,\theta} \in \mathbb{Z}^{2m\times m}$, $[\mathbf{D}|\mathbf{D}_\rho] \cdot \mathbf{E}_{1,\theta} = \mathbf{A}_\rho - \mathbf{V}_\theta$. Set $\mathsf{sk}_2^\rho = (\theta, \mathbf{E}_{1,\theta})_{\theta \in Path(\rho)}$.
4. Output the punctured secret key $\mathsf{sk}_{\mathsf{N}}^\rho = \{\mathsf{sk}_{1,\mathsf{N}}^\rho, \mathsf{sk}_2^\rho\}$.

$\mathsf{Punc}(\mathsf{pk}, \mathsf{sk}_{\mathsf{N}}^\rho, \eta)$: On input the public key pk, a secret key $\mathsf{sk}_{\mathsf{N}}^\rho = \{\mathsf{sk}_{1,\mathsf{N}}^\rho, \mathsf{sk}_2^\rho\}$, where $\mathsf{sk}_{1,\mathsf{N}}^\rho = \mathbf{T}_{\mathsf{N}}^\rho$ is punctured on a positive tag ρ and the negative tag set $\mathsf{N} = \{\eta_1, \cdots, \eta_\ell\}$, and a negative tag $\eta \in \mathcal{N}$, do as follows:

1. Evaluate $\mathbf{B}_{f_\eta} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\{\mathbf{B}_i\}_{i=1}^d, f_\eta)$.
2. Compute $\mathbf{T}_{\mathsf{N}\cup\{\eta\}}^{\rho,\mathrm{ER}} \leftarrow \mathsf{ExtendRight}\left(\left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_\ell}}\right], \mathbf{T}_{\mathsf{N}}^\rho, \mathbf{B}_{f_\eta}\right)$.
3. Finally, compute the re-randomized trapdoor as $\mathbf{T}_{\mathsf{N}\cup\{\eta\}}^\rho \leftarrow \mathsf{RandBasis}\left(\left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_\ell}}\middle|\mathbf{B}_{f_\eta}\right], \mathbf{T}_{\mathsf{N}\cup\{\eta\}}^{\rho,\mathrm{ER}}, \sigma_{\ell+2}\right)$, where $\sigma_{\ell+2} = \sigma_0 \cdot (\sqrt{m \log m})^{\ell+1}$. Set $\mathsf{sk}_{1,\mathsf{N}\cup\{\eta\}}^\rho = \mathbf{T}_{\mathsf{N}\cup\{\eta\}}^\rho \in \mathbb{Z}_q^{(\ell+3)m\times(\ell+3)m}$.
4. Output the punctured secret key $\mathsf{sk}_{\mathsf{N}\cup\{\eta\}}^\rho = \{\mathsf{sk}_{1,\mathsf{N}\cup\{\eta\}}^\rho, \mathsf{sk}_2^\rho\}$.

$\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}, \rho, \boldsymbol{\eta}, t)$: On input the public key pk, a message $\mathbf{m} \in \{0,1\}^m$, a positive tag $\rho \in \mathbb{Z}_q^n$, a d-tuple of negative tags $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d) \in \mathbb{Z}_q^d$, where each $\eta_i \in \mathbb{Z}_q$, and a time period $t \in \mathcal{T}$, do as follows:

1. Choose a uniformly random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, error vectors $\mathbf{e}_0, \mathbf{e}_{out}, \mathbf{e}'_0 \in \chi^m$; and $d+3$ uniformly random matrices $\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_d, \mathbf{S}_1, \mathbf{S}_2 \in \{+1, -1\}^{m \times m}$.
2. Evaluate the encoding of ρ, t as $\mathbf{H}_\rho, \mathbf{H}_t$ and set $\mathbf{A}_t = \mathbf{A}_1 + \mathbf{H}_t\mathbf{G}$, $\mathbf{D}_\rho = \mathbf{D}_0 + \mathbf{H}_\rho\mathbf{G}$, $\mathbf{D}_t = \mathbf{D}_1 + \mathbf{H}_t\mathbf{G}$.
3. Set $\mathbf{H}_1 = \left[\mathbf{A} \middle| \mathbf{A}_t \middle| \eta_1\mathbf{G} + \mathbf{B}_1 \middle| \cdots \middle| \eta_d\mathbf{G} + \mathbf{B}_d\right] \in \mathbb{Z}_q^{n \times (d+2)m}$, and an error vector $\mathbf{e} = \left[\mathbf{I}_m \middle| \mathbf{R}_0 \middle| \mathbf{R}_1 \middle| \cdots \middle| \mathbf{R}_d\right]^\top \cdot \mathbf{e}_0$.
   Compute $\mathbf{c} = \left[\mathbf{c}_{in} \middle| \bar{\mathbf{c}}_1 \middle| \mathbf{c}_1 \middle| \cdots \middle| \mathbf{c}_d\right] = \mathbf{H}_1^\top\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^{(d+2)m}$, where $\mathbf{c}_{in} = \mathbf{A}^\top\mathbf{s} + \mathbf{e}_0$, $\bar{\mathbf{c}}_1 = \mathbf{A}_t^\top\mathbf{s} + \mathbf{R}_0^\top\mathbf{e}_0$, and $\mathbf{c}_i = (\eta_i\mathbf{G} + \mathbf{B}_i)^\top\mathbf{s} + \mathbf{R}_i^\top\mathbf{e}_0 \ \forall\ i \in \{1, \cdots, d\}$.
4. Compute $\mathbf{c}_{out} = \mathbf{U}^\top\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \cdot \mathbf{m} \in \mathbb{Z}_q^m$.
5. Set $\mathbf{H}_2 = \left[\mathbf{D} \middle| \mathbf{D}_\rho \middle| \mathbf{D}_t\right] \in \mathbb{Z}_q^{n \times 3m}$ and an error vector $\mathbf{e}' = \left[\mathbf{I}_m \middle| \mathbf{S}_1 \middle| \mathbf{S}_2\right]^\top \cdot \mathbf{e}'_0$. Compute $\tilde{\mathbf{c}} = [\tilde{\mathbf{c}}_0 | \tilde{\mathbf{c}}_1 | \tilde{\mathbf{c}}_2] = \mathbf{H}_2^\top\mathbf{s} + \mathbf{e}' \in \mathbb{Z}_q^{3m}$, where $\tilde{\mathbf{c}}_0 = \mathbf{D}^\top\mathbf{s} + \mathbf{e}'_0$, $\tilde{\mathbf{c}}_1 = \mathbf{D}_\rho^\top\mathbf{s} + \mathbf{S}_1^\top\mathbf{e}'_0$, and $\tilde{\mathbf{c}}_2 = \mathbf{D}_t^\top\mathbf{s} + \mathbf{S}_2^\top\mathbf{e}'_0$.
6. Output the ciphertext $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}}) \in \mathbb{Z}_q^{(d+6)m}$.

$\mathsf{RevAccK}(\mathsf{pk}, \mathsf{msk}, \mathsf{P}, t)$: On input the public key pk, the master secret key msk, a time period t, a set of revoked positive tags P on time t, do as follows:

1. Compute $\mathbf{D}_t = \mathbf{D}_1 + \mathbf{H}_t\mathbf{G} \in \mathbb{Z}_q^{n \times m}$.
2. For each $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{P})$, retrieve $\mathbf{V}_\theta$ (which is always pre-defined in algorithm $\mathbf{KeyGen}$), and sample $\mathbf{E}_{2,\theta} \leftarrow \mathsf{SampleRight}\,(\mathbf{D}, \mathbf{T}_\mathbf{D}, \mathbf{D}_t, \mathbf{V}_\theta, \sigma_0)$. Note that $\mathbf{E}_{2,\theta} \in \mathbb{Z}^{2m \times m}$ and $[\mathbf{D}|\mathbf{D}_t] \cdot \mathbf{E}_{2,\theta} = \mathbf{V}_\theta$.
3. Output the revoke access key for time t as $\mathsf{rk}_\mathsf{P}^t = (\theta, \mathbf{E}_{2,\theta})_{\theta \in \mathsf{KUNodes}(\mathsf{BT},\mathsf{P})}$.

$\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_\mathsf{N}^{\rho'}, \mathsf{rk}_\mathsf{P}^t, ct, \rho, \boldsymbol{\eta}, t)$: On input the public key pk, the punctured secret key $\mathsf{sk}_\mathsf{N}^{\rho'} = \{\mathsf{sk}_{1,\mathsf{N}}^{\rho'}, \mathsf{sk}_2^{\rho'}\}$, where $\mathsf{sk}_{1,\mathsf{N}}^{\rho'} = \mathbf{T}_\mathsf{N}^{\rho'}$, punctured on the negative tag set $\mathsf{N} = \{\eta'_1, \cdots, \eta'_\ell\}$, a positive tag ρ' and $\mathsf{sk}_2^{\rho'} = (\theta, \mathbf{E}_{1,\theta})_{\theta \in I)}$, a revoke access key $\mathsf{rk}_\mathsf{P}^t = (\theta, \mathbf{E}_{2,\theta})_{\theta \in J}$ for some set of nodes I, J, for time t and a ciphertext $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$ with a positive tag $\rho \in \mathbb{Z}_q^n$, a d-tuple of negative tags $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d) \in \mathbb{Z}_q^d$, and a time period $t \in \mathcal{T}$, do as follows:

1. If $I \cap J = \emptyset$, or $\rho \neq \rho'$, or $\exists$ some $j \in \{1, \cdots, \ell\}$ such that $f_{\eta'_j}(\boldsymbol{\eta}) \neq 0$, outputs $\perp$. Otherwise, proceed as below:
   - As $I \cap J \neq \phi$, choose $\theta \in I \cap J$, and set $\mathbf{E}_1 = \mathbf{E}_{1,\theta}$, and $\mathbf{E}_2 = \mathbf{E}_{2,\theta}$. Note that $[\mathbf{D}|\mathbf{D}_\rho] \cdot \mathbf{E}_1 + [\mathbf{D}|\mathbf{D}_t] \cdot \mathbf{E}_2 = \mathbf{A}_\rho$.
   - Parse $\tilde{\mathbf{c}} = [\tilde{\mathbf{c}}_0 | \tilde{\mathbf{c}}_1 | \tilde{\mathbf{c}}_2]$, where $\tilde{\mathbf{c}}_i \in \mathbb{Z}_q^m$, for $i \in \{0, 1, 2\}$. Compute $\bar{\mathbf{c}}_0 = \mathbf{E}_1^\top[\tilde{\mathbf{c}}_0 | \tilde{\mathbf{c}}_1] + \mathbf{E}_2^\top[\tilde{\mathbf{c}}_0 | \tilde{\mathbf{c}}_2] \in \mathbb{Z}_q^m$.
   - Evaluate the encoding of ρ, t as $\mathbf{H}_\rho, \mathbf{H}_t$ and set $\mathbf{A}_t = \mathbf{A}_1 + \mathbf{H}_t\mathbf{G}$, $\mathbf{A}_\rho = \mathbf{A}_0 + \mathbf{H}_\rho\mathbf{G}$. Set $\mathbf{H}' = \left[\mathbf{A} \middle| \mathbf{A}_\rho \middle| \mathbf{A}_t \middle| \eta_1\mathbf{G} + \mathbf{B}_1 \middle| \cdots \middle| \eta_d\mathbf{G} + \mathbf{B}_d\right]$.

- Since $\rho' = \rho$, $\mathbf{T}_{\mathsf{N}}^{\rho'} = \mathbf{T}_{\mathsf{N}}^{\rho}$ is a trapdoor for $\left[\mathbf{A}\middle|\mathbf{A}_{\rho}\middle|\mathbf{B}_{f_{\eta'_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta'_\ell}}\right]$. Compute $\mathbf{T}_{\mathsf{N}}^{\rho,\mathrm{ER}} \leftarrow \mathsf{ExtendRight}\left(\left[\mathbf{A}\middle|\mathbf{A}_{\rho}\middle|\mathbf{B}_{f_{\eta'_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta'_\ell}}\right], \mathbf{T}_{\mathsf{N}}^{\rho}, \mathbf{A}_t\right)$. We switch the rows of $\mathbf{T}_{\mathsf{N}}^{\rho,\mathrm{ER}}$ to get a matrix $\mathbf{T}_{\mathbf{H}'}$, a trapdoor for $\mathbf{H}'$. This operation does not change the Gram-Schmidt norm of the basis.
- Sample $\mathbf{R} \leftarrow \mathsf{SampleD}\left(\left[\mathbf{A}\middle|\mathbf{A}_{\rho}\middle|\mathbf{A}_t\middle|\mathbf{B}_{f_{\eta'_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta'_\ell}}\right], \mathbf{T}_{\mathbf{H}'}, \mathbf{U}, \sigma_{\ell+1}\right)$.
- Parse $\mathbf{c} = \left[\mathbf{c}_{in}\middle|\bar{\mathbf{c}}_1\middle|\mathbf{c}_1\middle|\cdots\middle|\mathbf{c}_d\right]$. Evaluate $c_{f_{\eta'_j}} \leftarrow \mathsf{Eval}_{ct}(\{\eta'_i, \mathbf{B}_i, \mathbf{c}_i\}_{i=1}^{d}, f_{\eta'_j})$ for each $j \in \{1, \cdots, \ell\}$. Set $\mathbf{c}' = \left[\mathbf{c}_{in}\middle|\bar{\mathbf{c}}_0\middle|\bar{\mathbf{c}}_1\middle|\mathbf{c}_{f_{\eta'_1}}\middle|\cdots\middle|\mathbf{c}_{f_{\eta'_\ell}}\right]$.
- Compute $(m_1, \cdots, m_m) = \mathbf{c}_{out} - \mathbf{R}^{\top}\mathbf{c}'$. For each i, if $|m_i| < q/4$, take $m_i = 0$, otherwise take $m_i = 1$. Output $\mathbf{m} = (m_1, \cdots, m_m)$.

$\mathsf{Token}(\mathsf{pk}, \mathsf{msk}, \mathcal{E}_1, \mathcal{E}_2)$: On input the public key pk, the master secret key msk, and two tuples $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1)$, $\mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2(> t_1))$ consisting of a positive tag, a d-tuple of negative tags and a time, do as follows:

1. Let $\mathbf{A}_{\rho_1} = \mathbf{A}_0 + \mathbf{H}_{\rho_1}\mathbf{G}$, $\mathbf{A}_{\rho_2} = \mathbf{A}_0 + \mathbf{H}_{\rho_2}\mathbf{G}$, $\mathbf{A}_{t_1} = \mathbf{A}_1 + \mathbf{H}_{t_1}\mathbf{G}, \mathbf{A}_{t_2} = \mathbf{A}_1 + \mathbf{H}_{t_2}\mathbf{G}, \mathbf{D}_{\rho_1} = \mathbf{D}_0 + \mathbf{H}_{\rho_1}\mathbf{G}$, $\mathbf{D}_{\rho_2} = \mathbf{D}_0 + \mathbf{H}_{\rho_2}\mathbf{G}$, $\mathbf{D}_{t_1} = \mathbf{D}_1 + \mathbf{H}_{t_1}\mathbf{G}, \mathbf{D}_{t_2} = \mathbf{D}_1 + \mathbf{H}_{t_2}\mathbf{G}$, where $\mathbf{H}_{\rho_1}, \mathbf{H}_{\rho_2}, \mathbf{H}_{t_1}, \mathbf{H}_{t_2}$ are the encoding of ρ_1, ρ_2, t_1, t_2, respectively. Let $\boldsymbol{\eta}_1 = (\eta_1, \cdots, \eta_d), \boldsymbol{\eta}_2 = (\eta'_1, \cdots, \eta'_d)$.
2. If $\boldsymbol{\eta}_1 = \boldsymbol{\eta}_2$, proceed as follows:
   - Let $\mathsf{L} = \left[\eta_1\mathbf{G} + \mathbf{B}_1\middle|\cdots\middle|\eta_d\mathbf{G} + \mathbf{B}_d\right] \in \mathbb{Z}_q^{n \times dm}$.
   - Choose $\mathbf{Y} \leftarrow \chi^{m \times m}$. Sample $\mathbf{X} \leftarrow \mathsf{SampleRight}\left(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathsf{L}, \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}, \sigma_0\right)$.
     $$\begin{cases} \text{Note that } \mathbf{X} \in \mathbb{Z}^{(d+1)m \times m} \text{ and } \left[\mathbf{A}\middle|\mathsf{L}\right] \cdot \mathbf{X} = \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}; \\ \text{Parse } \mathbf{X} \text{ as } (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}) \in \mathbb{Z}^{m \times m} \times \mathbb{Z}^{dm \times m}; \\ \text{Observe that } \mathbf{A} \cdot \mathbf{X}^{(0)} + \mathsf{L} \cdot \mathbf{X}^{(1)} = \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}. \end{cases}$$
   - Set $\Delta_1 = \begin{bmatrix} \mathbf{I}_{m \times m} & \mathbf{X}^{(0)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}^{(1)} & \mathbf{I}_{dm \times dm} \end{bmatrix} \in \mathbb{Z}^{(d+2)m \times (d+2)m}$.

   If $\boldsymbol{\eta}_1 \neq \boldsymbol{\eta}_2$, proceed as follows:
   - Without loss of generality, we assume that first $j(\leq d)$ components of $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$ are different. We have $\eta_1 \neq \eta'_1, \cdots, \eta_j \neq \eta'_j$, and $\eta_{j+1} = \eta'_{j+1}, \cdots, \eta_d = \eta'_d$. Let $\mathsf{M} = \left[\eta_{j+1}\mathbf{G} + \mathbf{B}_{j+1}\middle|\cdots\middle|\eta_d\mathbf{G} + \mathbf{B}_d\right] \in \mathbb{Z}_q^{n \times (d-j)m}$.
   - Choose $\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_j \leftarrow \chi^{m \times m}$ and sample $\mathbf{X}_0 \leftarrow \mathsf{SampleRight}\left(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathsf{M}, \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}_0, \sigma_0\right)$ and $\forall\ i \in \{1, \cdots, j\}$, $\mathbf{X}_i \leftarrow \mathsf{SampleRight}\left(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathsf{M}, (\eta'_{d_i}\mathbf{G} + \mathbf{B}_{d_i}) - (\eta_{d_i}\mathbf{G} + \mathbf{B}_{d_i})\mathbf{Y}_i, \sigma_0\right)$.
     $$\begin{cases} \text{Note that } \mathbf{X}_0 \in \mathbb{Z}^{(d-j+1)m \times m} \text{ and } \left[\mathbf{A}\middle|\mathsf{M}\right] \cdot \mathbf{X}_0 = \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}_0; \\ \text{Parse } \mathbf{X}_0 \text{ as } (\mathbf{X}_0^{(0)}, \mathbf{X}_0^{(j+1)}, \cdots, \mathbf{X}_0^{(d)}) \in \underbrace{\mathbb{Z}^{m \times m} \times \cdots \times \mathbb{Z}^{m \times m}}_{(d-j+1) \text{ times}}; \\ \text{Observe that} \\ \mathbf{A} \cdot \mathbf{X}_0^{(0)} + (\eta_{j+1}\mathbf{G} + \mathbf{B}_{j+1}) \cdot \mathbf{X}_0^{(j+1)} + \cdots + (\eta_d\mathbf{G} + \mathbf{B}_d) \cdot \mathbf{X}_0^{(d)} \\ \qquad = \mathbf{A}_{t_2} - \mathbf{A}_{t_1}\mathbf{Y}. \end{cases}$$

For each $i = \{1, \cdots, j\}$, the following holds:

$$\begin{cases} \mathbf{X}_i \in \mathbb{Z}^{(d-j+1)m \times m} \text{ and } \left[\mathbf{A} \middle| \mathsf{M}\right] \cdot \mathbf{X}_i = (\eta'_{d_i}\mathbf{G} + \mathbf{B}_{d_i}) - (\eta_{d_i}\mathbf{G} + \mathbf{B}_{d_i})\mathbf{Y}_i; \\ \text{Parse } \mathbf{X}_i \text{ as } (\mathbf{X}_i^{(0)}, \mathbf{X}_i^{(j+1)}, \cdots, \mathbf{X}_i^{(d)}) \in \underbrace{\mathbb{Z}^{m\times m} \times \cdots \times \mathbb{Z}^{m\times m}}_{(d-j+1) \text{ times}}; \\ \text{Observe that} \\ \mathbf{A} \cdot \mathbf{X}_i^{(0)} + (\eta_{j+1}\mathbf{G} + \mathbf{B}_{j+1}) \cdot \mathbf{X}_i^{(j+1)} + \cdots + (\eta_d \mathbf{G} + \mathbf{B}_d) \cdot \mathbf{X}_0^{(d)} \\ \qquad = (\eta'_{d_i}\mathbf{G} + \mathbf{B}_{d_i}) - (\eta_{d_i}\mathbf{G} + \mathbf{B}_{d_i})\mathbf{Y}_i. \end{cases}$$

– Set

$$\Delta_1 = \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{X}_0^{(0)} & \mathbf{X}_1^{(0)} & \cdots & \mathbf{X}_j^{(0)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_0 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Y}_j & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_0^{(j+1)} & \mathbf{X}_1^{(j+1)} & \cdots & \mathbf{X}_j^{(j+1)} & \mathbf{I}_{m\times m} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{X}_0^{(d)} & \mathbf{X}_1^{(d)} & \cdots & \mathbf{X}_j^{(d)} & \mathbf{0} & \cdots & \mathbf{I}_{m\times m} \end{bmatrix}.$$

Using basic algebra, it is easy to form the matrix $\Delta_1 \in \mathbb{Z}^{(d+2)m\times(d+2)m}$ containing $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_j$ and $\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_j$, satisfying $\left[\mathbf{A}\middle|\mathbf{A}_{t_1}\middle|\eta_1\mathbf{G} + \mathbf{B}_1\middle|\cdots\middle|\eta_d\mathbf{G} + \mathbf{B}_d\right] \cdot \Delta_1 = \left[\mathbf{A}\middle|\mathbf{A}_{t_2}\middle|\eta'_1\mathbf{G} + \mathbf{B}_1\middle|\cdots\middle|\eta'_d\mathbf{G} + \mathbf{B}_d\right]$.

3. Let $\mathsf{N} = \left[\mathbf{A}_{t_1}\middle|\eta_1\mathbf{G} + \mathbf{B}_1\middle|\cdots\middle|\eta_d\mathbf{G} + \mathbf{B}_d\right] \in \mathbb{Z}_q^{n\times(d+1)m}$. If $\rho_1 = \rho_2$, proceed as follows:
   – Choose $\mathbf{Y}' \leftarrow \chi^{m\times m}$ and sample $\mathbf{X}' \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, [\mathsf{N}|\mathbf{D}_{\rho_1}], \mathbf{D}_{t_2} - \mathbf{D}_{t_1}\mathbf{Y}', \sigma_0)$.

$$\begin{cases} \text{Note that } \mathbf{X}' \in \mathbb{Z}^{(d+3)m\times m} \text{ and } \left[\mathbf{A}\middle|\mathsf{N}\middle|\mathbf{D}_{\rho_1}\right] \cdot \mathbf{X}' = \mathbf{D}_{t_2} - \mathbf{D}_{t_1}\mathbf{Y}'; \\ \text{Parse } \mathbf{X}' \text{ as } (\mathbf{X}'^{(0)}, \mathbf{X}'^{(1)}) \in \mathbb{Z}^{(d+2)m\times m} \times \mathbb{Z}^{m\times m}; \\ \text{Observe that } [\mathbf{A}|\mathsf{N}] \cdot \mathbf{X}'^{(0)} + \mathbf{D}_{\rho_1} \cdot \mathbf{X}'^{(1)} = \mathbf{D}_{t_2} - \mathbf{D}_{t_1}\mathbf{Y}'. \end{cases}$$

   – Set $\Delta_2 = \begin{bmatrix} \mathbf{0}_{(d+2)m\times m} & \mathbf{0}_{(d+2)m\times m} & \mathbf{X}'^{(0)} \\ \mathbf{I}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} & \mathbf{X}'^{(1)} \\ \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{Y}' \end{bmatrix} \in \mathbb{Z}^{(d+5)m\times 3m}$.

   If $\rho_1 \neq \rho_2$, proceed as follows:
   – Choose $\mathbf{Y}_1, \mathbf{Y}_2 \leftarrow \chi^{m\times m}$ and sample $\mathbf{X}_1 \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathsf{N}, \mathbf{D}_{\rho_2} - \mathbf{D}_{\rho_1}\mathbf{Y}_1, \sigma_0)$ and $\mathbf{X}_2 \leftarrow \mathsf{SampleRight}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathsf{N}, \mathbf{D}_{t'} - \mathbf{D}_t\mathbf{Y}_2, \sigma_0)$.

$$\begin{cases} \text{Note that } \mathbf{X}'_1, \mathbf{X}'_2 \in \mathbb{Z}^{(d+2)m\times m} \text{ and} \\ \left[\mathbf{A}\middle|\mathsf{N}\right] \cdot \mathbf{X}'_1 = \mathbf{D}_{\rho_2} - \mathbf{D}_{\rho_1}\mathbf{Y}'_1, \ \left[\mathbf{A}\middle|\mathsf{N}\right] \cdot \mathbf{X}'_2 = \mathbf{D}_{t_2} - \mathbf{D}_{t_1}\mathbf{Y}'_2. \end{cases}$$

   – Set $\Delta_2 = \begin{bmatrix} \mathbf{0}_{(d+2)m\times m} & \mathbf{X}'_1 & \mathbf{X}'_2 \\ \mathbf{I}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{Y}'_1 & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{Y}'_2 \end{bmatrix} \in \mathbb{Z}^{(d+5)m\times 3m}$.

4. Output the update token $\mathsf{tk}_{\mathcal{E}_1\to\mathcal{E}_2} = \{\Delta_1, \Delta_2\}$.

$\mathsf{UpdateCT}(\mathsf{pk}, \mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2}, ct, \rho_1, \boldsymbol{\eta}_1, t_1)$: On input the public key pk, a ciphertext ct with tags ρ_1, $\boldsymbol{\eta}_1$ and a token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2} = \{\Delta_1, \Delta_2\}$ for $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1)$, $\mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2 (> t_1))$, do as follows:

1. Parse $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$, where $\mathbf{c} = [\mathbf{c}_{in} | \bar{\mathbf{c}}_1 | \mathbf{c}_1 | \cdots | \mathbf{c}_d]$, $\tilde{\mathbf{c}} = [\tilde{\mathbf{c}}_0 | \tilde{\mathbf{c}}_1 | \tilde{\mathbf{c}}_2]$.
2. Choose a uniformly random vector $\mathbf{r} \leftarrow \mathbb{Z}_q^n$ and error vectors $\tilde{\mathbf{e}}_1 \in \chi^{(d+2)m}$, $\tilde{\mathbf{e}}_2 \in \chi^m, \tilde{\mathbf{e}}_3 \in \chi^{3m}$.
3. Let $\boldsymbol{\eta}_2 = (\eta'_1, \cdots, \eta'_d)$. Set $\mathbf{H}'_1 = \big[\mathbf{A} \big| \mathbf{A}_{t_2} \big| \eta'_1\mathbf{G} + \mathbf{B}_1 \big| \cdots \big| \eta'_d\mathbf{G} + \mathbf{B}_d\big]$ and $\mathbf{H}'_2 = \big[\mathbf{D} \big| \mathbf{D}_{\rho_2} \big| \mathbf{D}_{t_2}\mathbf{G}\big]$.
4. Compute $ct' = (\mathbf{c}', \mathbf{c}'_{out}, \tilde{\mathbf{c}}')$ as $\mathbf{c}' = \Delta_1^\top \cdot \mathbf{c} + \mathbf{H}_1'^\top \mathbf{r} + \tilde{\mathbf{e}}_1$, $\mathbf{c}'_{out} = \mathbf{c}_{out} + \mathbf{U}^\top \mathbf{r} + \tilde{\mathbf{e}}_2$, and $\tilde{\mathbf{c}}' = \Delta_2^\top \cdot [\mathbf{c} | \tilde{\mathbf{c}}] + \mathbf{H}_2'^\top \mathbf{r} + \tilde{\mathbf{e}}_3$.
5. Output the updated ciphertext $ct' \in \mathbb{Z}_q^{(d+6)m}$.

5 Analysis of the proposed construction

In this section, we analyzed the correctness, efficiency, and securityof the proposed construction. Due to page restrictions, we provide the proof sketches here, and the full proofs of correctness and security are deferred to Appendix B and Appendix C, respectively.

Theorem 1 (Correctness). *The* DPE *scheme is correct if the following condition holds:* $3\alpha_{\mathcal{F}}^3 \cdot \chi_{max}^2 \cdot (d+3)^2(\ell+3)^2 \cdot m^{\frac{\ell+1}{2}+3} < q/4$.

Proof sketch. We argue that Dec outputs the correct message with all but negligible probability for both honestly generated and correctly updated ciphertexts. Let $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$ be an honestly computed ciphertext of $\mathbf{m}$ under positive tag ρ, negative tag set $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_d\}$ on time t. If the user's access has not been revoked in either of the two revocation layers, then by using its secret key $\mathsf{sk}_{\mathsf{N}}^{\rho}$ and the public revoke-access key $\mathsf{rk}_{\mathsf{P}}^{t}$, the user computes $\bar{\mathbf{c}}_0 = \mathbf{A}_\rho^\top \mathbf{s} + \mathsf{error}_1$. Moreover, for every punctured negative tag, evaluation yields $\mathbf{c}_{f_{\eta_j}} = \mathbf{B}_{f_{\eta_j}}^\top \mathbf{s} + \mathbf{e}$ with $\|\mathbf{e}\| < \Delta$. Finally, it evaluates $\mathbf{c}_{out} - \mathbf{R}^\top [\mathbf{c}_{in} \mid \bar{\mathbf{c}}_0 \mid \cdots | \mathbf{c}_{f_{\eta_\ell}}] = \lfloor q/2 \rfloor \cdot \mathbf{m} + \mathsf{error}_2$, where error_2 aggregates error_1, and some other noises. By Lemma 1 and the tail bounds of the discrete Gaussian sampling, $\|\mathsf{error}_1\|$ is small with overwhelming probability. Using the norm bounds on $\mathbf{R}$ (by Lemma 1), error_1 and the other noises, we obtain $\|\mathsf{error}_2\| \leq \beta_1$ with overwhelming probability; choosing parameters so that $\beta_1 < q/4$ implies correct output of the original message $\mathbf{m}$.

Let $ct' = (\mathbf{c}', \mathbf{c}'_{out}, \tilde{\mathbf{c}}')$ be an updated ciphertexts obtained from ct using a valid token $\mathsf{tk}_{\mathcal{E}_1 \to \mathcal{E}_2} = (\Delta_1, \Delta_2)$ for $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1)$ and $\mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2)$ with $t_2 > t_1$. This update preserves the message and maintains consistency of the shared randomness by effectively shifting $\mathbf{s}$ to $\mathbf{s} + \mathbf{r}$ for uniform $\mathbf{r} \leftarrow \mathbb{Z}_q^n$, i.e., $\mathbf{c}' = \mathbf{H}_1'^\top(\mathbf{s}+\mathbf{r}) + \text{noise}$, $\tilde{\mathbf{c}}' = \mathbf{H}_2'^\top(\mathbf{s}+\mathbf{r}) + \text{noise}$, and $\mathbf{c}'_{out} = \mathbf{U}^\top(\mathbf{s}+\mathbf{r}) + \lfloor q/2 \rfloor \mathbf{m} + \text{noise}$. If $\rho_2 \notin \mathsf{P}_{t_2}$, the same transform-and-decrypt procedure yields $\lfloor q/2 \rfloor \mathbf{m} + \mathsf{error}'_2$. Bounding $\|\Delta_1\|, \|\Delta_2\|$ (by Lemma 1 and the structure of $\mathsf{SampleRight}$) and the additional update noises gives $\|\mathsf{error}'_2\| \leq \beta_2$ with overwhelming probability. Hence, for $\beta = \max\{\beta_1, \beta_2\} < q/4$, decryption is correct for both original and updated ciphertexts. We set $\beta = 3\alpha_{\mathcal{F}}^3 \cdot \chi_{max}^2 \cdot (d+3)^2(\ell+3)^2 \cdot m^{\frac{\ell+1}{2}+3} < q/4$.

Efficiency
Table 2 summarizes the asymptotic bit-size of public parameter, secret key, revoke access key, ciphertext, token, and updated ciphertext in DPE scheme.

Table 2. Data Sizes of Proposed DPE

Public parameter size	$O(d \cdot n^2 \log^2 q)$
Secret key size	$O(\eta^2 \cdot n^2 \log^3 q)$
punctured (with tags $\{t_1, \cdots, t_\eta\}$)	
Revoke access Key size	$O(r \log(N/r) \cdot n^2 \log^2 q)$
Ciphertext size	$O(d \cdot n \log^2 q))$
Token size	$O(d^2 \cdot n^2 \log^3 q)$
Updated Ciphertext size	$O(d \cdot n \log^2 q))$

d is the maximum number of negative tags per ciphertext; N is the maximal number of positive tags the system will support; ℓ is the punctured times; above secret key sizes and ciphertext size, updated ciphertext size are given with respect to the maximum number of negative tags per ciphertext d; r is the number of revoked positive tags; n is an integer; q is a prime.

Theorem 2 (Security). *The above scheme is* IND-sDPE-CPA *secure assuming the hardness of* $\mathsf{dLWE}_{n,3m,q,\chi}$.

Proof sketch. We prove IND-sDPE-CPA security via a sequence of games. The adversary $\mathcal{A}$ commits upfront to the challenge tuple $(\rho^*, \boldsymbol{\eta}^*, t^*)$, and the challenger answers the puncture, corruption, revoke-access key queries, and token, encryption, update queries, subject to the experiment's restrictions. Game 0 is the original IND-sDPE-CPA experiment. In Game 1, we differently set $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d$, $\mathbf{D}_0, \mathbf{D}_1$ from the public key pk to embed the challenge tags and time $(\rho^*, \boldsymbol{\eta}^*, t^*)$. The remainder of this game is unchanged. Game 0 is statistically indistinguishable from Game 1 due to the left-over hash lemma. In Game 2, we change how $\mathbf{A}, \mathbf{D}$ in pk are chosen. Here, the challenger chooses random $\mathbf{A}, \mathbf{D}$ from $\mathbb{Z}_q^{n \times m}$ instead of using the TrapGen algorithm. We then simulate all key and revocation oracles using trapdoor delegation and sampling techniques. For example, in the real scheme, the keys $\mathsf{sk}^{\rho}_{1,\mathsf{N}_\rho}$ are generated using the algorithm ExtendRight, RandBasis for every ρ. In contrast, during the simulation, they are drawn from ExtendLeft, ExtendRight, RandBasis. The properties of these sampling algorithms (see Section 3) ensure that the resulting distributions are statistically indistinguishable. In particular, tokens for non-challenge tuples are generated using the trapdoor, while for the challenge tuple, we return tokens sampled from the correct distribution without revealing any trapdoor information, and updates are only permitted in a one-way manner. Besides, the way we embed the challenge tags and time in the public parameter, we can respond to the queries using the trapdoor $\mathbf{T}_\mathbf{G}$. Game 2 is otherwise same as Game 1. Since the keys and responses to the queries are statistically close to those in Game 1, Game 2 and Game 1 are statistically indistinguishable. Game 3 is identical to Game 2 except that the challenge ciphertext ct^* is chosen randomly from $\mathbb{Z}_q^{(d+6)m}$. Finally, we show that Game 2 and Game 3 are computationally indistinguishable for a PPT adversary, by giving a reduction from the dLWE problem.

Acknowledgements

This work is supported by Australian Research Council (ARC) Discovery Project under Grant DP220100003. The work of Willy Susilo is supported by the ARC Laureate Fellowship under Grant FL230100033. The work of Fuchun Guo is partially supported by ARC Future Fellowship under Grant FT220100046.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H) IBE in the standard model. In: EUROCRYPT 2010. pp. 553–572. Springer (2010)
2. Ajtai, M.: Generating hard instances of the short basis problem. In: ICALP 1999. pp. 1–9. Springer (1999)
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS 2009. pp. 75–86 (2009)
4. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In: EUROCRYPT 2014. pp. 533–556. Springer (2014)
5. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC 2013. pp. 575–584 (2013)
6. Brakerski, Z., Vaikuntanathan, V.: Circuit-abe from LWE: unbounded attributes and semi-adaptive security. In: CRYPTO 2016. pp. 363–384. Springer (2016)
7. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: EUROCRYPT 2010. pp. 523–552 (2010)
8. Derler, D., Jager, T., Slamanig, D., Striecks, C.: Bloom filter encryption and applications to efficient forward-secret 0-rtt key exchange. In: EUROCRYPT 2018. pp. 425–455. Springer (2018)
9. Derler, D., Krenn, S., Lorünser, T., Ramacher, S., Slamanig, D., Striecks, C.: Revisiting proxy re-encryption: Forward secrecy, improved security, and applications. In: PKC 2018. pp. 219–250. Springer (2018)
10. Derler, D., Ramacher, S., Slamanig, D., Striecks, C.: I want to forget: Fine-grained encryption with full forward secrecy in the distributed setting. IACR Cryptol. ePrint Arch. **2019**, 912 (2019)
11. Derler, D., Ramacher, S., Slamanig, D., Striecks, C.: Fine-grained forward secrecy: Allow-list/deny-list encryption and applications. In: International Conference on Financial Cryptography and Data Security. pp. 499–519. Springer (2021)
12. Dutta, P., Jiang, M., Duong, D.H., Susilo, W., Fukushima, K., Kiyomoto, S.: Hierarchical identity-based puncturable encryption from lattices with application to forward security. In: ACM AsiaCCS 2022. pp. 408–422 (2022)
13. Dutta, P., Susilo, W., Duong, D.H., Roy, P.S.: Puncturable identity-based encryption from lattices. In: ACISP 2021. pp. 571–589. Springer (2021)
14. Dutta, P., Susilo, W., Duong, D.H., Roy, P.S.: Puncturable identity-based and attribute-based encryption from lattices. Theoretical Computer Science **929**, 18–38 (2022)
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206 (2008)
16. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: 2015 IEEE S&P. pp. 305–320. IEEE (2015)

17. Günther, F., Hale, B., Jager, T., Lauer, S.: 0-rtt key exchange with full forward secrecy. In: EUROCRYPT 2017. pp. 519–548. Springer (2017)
18. Liu, H., Ming, Y., Wang, C., Zhao, Y.: Flexible selective data sharing with fine-grained erasure in vanets. IEEE TIFS (2024)
19. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012. pp. 700–718. Springer (2012)
20. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: CRYPTO 2001. pp. 41–62. Springer (2001)
21. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009. pp. 333–342 (2009)
22. Phuong, T.V.X., Ning, R., Xin, C., Wu, H.: Puncturable attribute-based encryption for secure data delivery in internet of things. In: INFOCOM 2018. pp. 1511–1519. IEEE (2018)
23. Phuong, T.V.X., Susilo, W., Kim, J., Yang, G., Liu, D.: Puncturable proxy re-encryption supporting to group messaging service. In: ESORICS 2019. pp. 215–233. Springer (2019)
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93 (2005)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)
26. Wei, J., Chen, X., Wang, J., Hu, X., Ma, J.: Forward-secure puncturable identity-based encryption for securing cloud emails. In: ESORICS 2019. pp. 134–150. Springer (2019)

A Deferred Prerequisites

Lemma 6 ([19]). $\mathbf{R}' \leftarrow \mathsf{DelTrap}(\mathbf{A}' = \left[\mathbf{A}\middle|\mathbf{A}_1\right], \mathbf{R}, \mathbf{H}', \mathbf{s})$*: On input an oracle* $\mathcal{O}$ *for discrete Gaussian sampling over cosets of* $\Lambda = \Lambda^{\perp}(\mathbf{A})$ *with parameter* $s \geq \eta_{\epsilon}(\Lambda^{\perp}(\mathbf{A}))$*, an extended matrix* $\mathbf{A}'$ *of* $\mathbf{A}$*, an invertible matrix* $\mathbf{H}'$*, the algorithm will sample (using* $\mathcal{O}$*) each column of* $\mathbf{R}'$ *independently from a discrete Gaussian with parameter* s *over the appropriate coset of* $\Lambda^{\perp}(\mathbf{A})$*, so that* $\mathbf{AR}' = \mathbf{H}'\mathbf{G} - \mathbf{A}_1$*. The algorithm outputs a trapdoor* $\mathbf{R}'$ *for* $\mathbf{A}'$ *with tag* $\mathbf{H}'$*.*

Lemma 7 ([19]). *Let* $q \geq 2, \overline{m} \geq 1$*,* $k = \lceil \log_2 q \rceil$*, and* $m = \overline{m} + nk = O(n \log q)$*.* $(\mathbf{s}, \mathbf{e}) \leftarrow \mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t)$ *: On input a uniform matrix* $\mathbf{A}$ *and its* $\mathbf{G}$*-trapdoor* $\mathbf{R}$*, and a vector* $\mathbf{b}$ *such that* $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$*,* Invert *returns* ($\mathbf{s}$ *and* $\mathbf{e}$)*. Note that if* $\boldsymbol{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$ *and* $1/\alpha \geq 2\sqrt{5(s_1(\mathbf{R})^2 + 1)} \cdot \omega(\sqrt{\log n})$ *then* Invert *succeeds with overwhelming probability over the choice of* $\mathbf{e}$*.*

Lemma 8 (Left-over Hash Lemma ([1], Lemma 13**)).** *Suppose that* $m > (n+1)\log_2 q + \omega(\log n)$ *and that* $q > 2$ *is prime. Let* $\mathbf{R}$ *be an* $m \times k$ *matrix chosen uniformly in* $\{1, -1\}^{m \times k} \mod q$*, where* $k = k(n)$ *is polynomial in* n*. Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be matrices chosen uniformly in* $\mathbb{Z}^{n \times m}$ *and* $\mathbb{Z}^{n \times k}$ *respectively. Then, for all vectors* $\mathbf{e} \in \mathbb{Z}_q^m$*, the distribution* $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^{\top}\mathbf{e})$ *is statistically close to the distribution* $(\mathbf{A}, \mathbf{B}, \mathbf{R}^{\top}\mathbf{e})$*.*

B Proof of Theorem 1

Proof. We will demonstrate that the scheme satisfies the correctness requirements with all but negligible probability. Suppose that $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$ is an honestly computed ciphertext of message $\mathbf{m}$ with respect to a positive tag ρ, a negative tag set $\boldsymbol{\eta} = \{\eta_1, \cdots, \eta_d\}$ on time t. Let the secret key for the positive tag ρ is $\mathsf{sk}_{\mathsf{N}}^{\rho} = \{\mathsf{sk}_{1,\mathsf{N}}^{\rho}, \mathsf{sk}_2^{\rho}\}$, where $\mathsf{sk}_{1,\mathsf{N}}^{\rho} = \mathbf{T}_{\mathsf{N}}^{\rho}$ on negative tag set $\mathsf{N} = \{\eta_1, \cdots, \eta_\ell\}$, a positive tag ρ and $\mathsf{sk}_1^{\rho} = (\theta, \mathbf{E}_{1,\theta})_{\theta \in I)}$. If ρ is not revoked at time t, then from secret key $\mathsf{sk}_{\mathsf{N}}^{\rho}$ and the revoke access key $\mathsf{rk}_{\mathsf{P}}^{t}$ on time t, we have $(\mathbf{E}_{1,\theta}, \mathbf{E}_{2,\theta})$, which satisfies $[\mathbf{D}|\mathbf{D}_\rho] \cdot \mathbf{E}_{1,\theta} + [\mathbf{D}|\mathbf{D}_t] \cdot \mathbf{E}_{2,\theta} = \mathbf{A}_\rho$.

To show that the decryption algorithm outputs the correct $\mathbf{m}$, it is required for Eval_{ct} that for $f_{\eta_j}(\boldsymbol{\eta}) = 0$, the resulting ciphertext $\mathbf{c}_{f_{\eta_j}} \in \mathbf{E}_{\mathbf{s},\Delta}(0, \mathbf{B}_{f_{\eta_j}})$ for all $j \in \{1, \cdots, \ell\}$. So that $\mathbf{c}_{f_{\eta_j}} = \mathbf{B}_{f_{\eta_j}}^{\top}\mathbf{s} + \mathbf{e}$ with $\|\mathbf{e}\| < \Delta < \chi_{max} \cdot \alpha_{\mathcal{F}}$.

Parse $\tilde{\mathbf{c}} = [\tilde{\mathbf{c}}_0|\tilde{\mathbf{c}}_1|\tilde{\mathbf{c}}_2]$, where $\tilde{c}_i \in \mathbb{Z}_q^m$, for $i \in \{0, 1, 2\}$. Compute

$$\begin{aligned}
\bar{\mathbf{c}}_0 &= \mathbf{E}_{1,\theta}^{\top}[\tilde{\mathbf{c}}_0|\tilde{\mathbf{c}}_1] + \mathbf{E}_{2,\theta}^{\top}[\tilde{\mathbf{c}}_0|\tilde{\mathbf{c}}_2] \\
&= \mathbf{E}_{1,\theta}^{\top}\left([\mathbf{D}|\mathbf{D}_\rho]^{\top}\mathbf{s} + \begin{bmatrix} \mathbf{e}_0' \\ \mathbf{S}_1^{\top}\mathbf{e}_0' \end{bmatrix}\right) + \mathbf{E}_{2,\theta}^{\top}\left([\mathbf{D}|\mathbf{D}_t]^{\top}\mathbf{s} + \begin{bmatrix} \mathbf{e}_0' \\ \mathbf{S}_2^{\top}\mathbf{e}_0' \end{bmatrix}\right) \\
&= \mathbf{A}_\rho^{\top}\mathbf{s} + \underbrace{\mathbf{E}_{1,\theta}^{\top}\begin{bmatrix} \mathbf{e}_0' \\ \mathbf{S}_1^{\top}\mathbf{e}_0' \end{bmatrix} + \mathbf{E}_{2,\theta}^{\top}\begin{bmatrix} \mathbf{e}_0' \\ \mathbf{S}_2^{\top}\mathbf{e}_0' \end{bmatrix}}_{\mathsf{error}_1}.
\end{aligned}$$

The following holds

$$\begin{aligned}
\mathbf{m} &= \mathbf{c}_{out} - \mathbf{R}^{\top}\left[\mathbf{c}_{in}\middle|\bar{\mathbf{c}}_0\middle|\bar{\mathbf{c}}_1\middle|\mathbf{c}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{c}_{f_{\eta_\ell}}\right] \\
&= \mathbf{c}_{out} - \mathbf{R}^{\top}\left(\left[\mathbf{A}|\mathbf{A}_\rho|\mathbf{A}_t|\cdots|\mathbf{B}_{f_{\eta_\ell}}\right]^{\top}\mathbf{s} + \left[\mathbf{e}_0|\ \ \mathsf{error}_1\ |\mathbf{R}_0^{\top}\mathbf{e}_0|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right]\right) \\
&= \mathbf{U}^{\top}\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \cdot \mathbf{m} - \mathbf{U}^{\top}\mathbf{s} - \mathbf{R}^{\top}\left[\mathbf{e}_0|\ \ \mathsf{error}_1\ |\mathbf{R}_0^{\top}\mathbf{e}_0|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right] \\
&= \lfloor q/2 \rfloor \cdot \mathbf{m} + \underbrace{\mathbf{e}_{out} - \mathbf{R}^{\top}\left[\mathbf{e}_0|\ \ \mathsf{error}_1\ |\mathbf{R}_0^{\top}\mathbf{e}_0|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right]}_{\mathsf{error}_2}.
\end{aligned}$$

To get a correct decryption of the original ciphertext ct, the norm of the above error term "error_2" should be less than $q/4$, i.e. it requires to satisfy $\left\|\mathbf{e}_{out} - \mathbf{R}^{\top} \cdot \left[\mathbf{e}_{in}|\mathbf{e}_0|\mathbf{e}_0'|\mathbf{e}_{f_{\eta_1}}|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right]\right\| < q/4$. Since, $\mathbf{E}_{1,\theta}, \mathbf{E}_{2,\theta} \in \mathbb{Z}^{2m \times m}$ are sampled from the distribution $\mathcal{D}_\sigma\big(\Lambda_q^{(\mathbf{A}_\rho - \mathbf{V}_\theta)}(\mathbf{D}|\mathbf{D}_\rho)\big)$, $\mathcal{D}_\sigma\big(\Lambda_q^{\mathbf{V}_\theta}(\mathbf{D}|\mathbf{D}_t)\big)$, respectively, we know that $[\mathbf{D}|\mathbf{D}_\rho] \cdot \mathbf{E}_{1,\theta} = \mathbf{A}_\rho - \mathbf{V}_\theta$, $[\mathbf{D}|\mathbf{D}_t] \cdot \mathbf{E}_{2,\theta} = \mathbf{V}_\theta$, and by Lemma 1, $\left\|\mathbf{E}_{1,\theta}^{\top}\right\|_2, \left\|\mathbf{E}_{2,\theta}^{\top}\right\|_2 < 2m\sigma$ with overwhelming probability. Since $\mathbf{S}_1, \mathbf{S}_2$ are random matrices in $\{+1, -1\}^{m \times m}$, by Lemma 1, $\left\|\mathbf{S}_1^{\top}\right\|_2, \left\|\mathbf{S}_2^{\top}\right\|_2 < 20\sqrt{m}$. Since $\mathbf{e}_0' \in \chi^m$, we know $\|\mathbf{e}_0'\| \leq \chi_{max}$. Finally, we have $\|\mathsf{error}_1\| < 2(1 + 20\sqrt{m})m\sigma\chi_{max}$ with overwhelming probability.

Since $\mathbf{e}_{out} \in \chi^m$, we know $\|\mathbf{e}_{out}\| \leq \chi_{max}$. Since $\mathbf{R} \in \mathbb{Z}^{(\ell+3)m \times n}$ is sampled from the distribution $\mathcal{D}_{\sigma_{\ell+1}}\big(\Lambda_q^{\mathbf{U}}(\mathbf{A}|\mathbf{A}_\rho|\mathbf{A}_t|\mathbf{B}_{f_{\eta_1'}}|\cdots|\mathbf{B}_{f_{\eta_\ell'}})\big)$, we know that $[\mathbf{A}|\mathbf{A}_\rho|\mathbf{A}_t|\mathbf{B}_{f_{\eta_1'}}|\cdots|\mathbf{B}_{f_{\eta_\ell'}}] \cdot \mathbf{R} = \mathbf{U}$ and by Lemma 1, $\left\|\mathbf{R}^{\top}\right\| < (\ell+3)m\sigma_{\ell+1}$ with overwhelming probability. Since $\mathbf{R}_0$ is random in $\{+1, -1\}^{m \times m}$, by Lemma 1,

we have that $\left\|\mathbf{R}_0^\top\right\|_2 < 20\sqrt{m}$. So we have $\left\|\left[\mathbf{e}_0|\ \mathsf{error}_1\ |\mathbf{R}_0^\top\mathbf{e}_0|\mathbf{e}_{f_{\eta_1}}|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right]\right\| < \chi_{max}+2(1+20\sqrt{m})m\sigma\chi_{max}+20\sqrt{m}\chi_{max}+\ell\Delta < (\ell\alpha_{\mathcal{F}}+m\sqrt{m}\sigma+\sqrt{m}+1)\chi_{max}$. Finally, using $\sigma_{\ell+1} = \sigma_0\cdot(\sqrt{m\log m})^{\ell+1}$, $\sigma = \omega(\alpha_{\mathcal{F}}\cdot\sqrt{\log m})$, we have $\|\mathsf{error}_2\| = \left\|\mathbf{e}_{out} - \mathbf{R}^\top\cdot\left[\mathbf{e}_0|\ \mathsf{error}_1\ |\mathbf{R}_0^\top\mathbf{e}_0|\mathbf{e}_{f_{\eta_1}}|\cdots|\mathbf{e}_{f_{\eta_\ell}}\right]\right\| \le \chi_{max} + (\ell+3)m\sigma_{\ell+1}\cdot(\ell\alpha_{\mathcal{F}} + m\sqrt{m}\sigma+\sqrt{m}+1)\chi_{max} \le 3\alpha_{\mathcal{F}}^2\cdot\chi_{max}\cdot(\ell+3)^2\cdot m^{\frac{\ell}{2}+3}$ with overwhelming probability. Let $\beta_1 = 3\alpha_{\mathcal{F}}^2\cdot\chi_{max}\cdot(\ell+3)^2\cdot m^{\frac{\ell}{2}+3}$. By choosing the parameters such that $\beta_1 < q/4$, the decryption will be correct.

To show the correct decryption of an updated ciphertext, we proceed as follows. Let $ct = (\mathbf{c}, \mathbf{c}_{out}, \tilde{\mathbf{c}})$ has been updated to $ct' = (\mathbf{c}', \mathbf{c}'_{out}, \tilde{\mathbf{c}}')$ with the update token $\mathsf{tk}_{\mathcal{E}_1\to\mathcal{E}_2} = \{\Delta_1, \Delta_2\}$, where $\mathcal{E}_1 = (\rho_1, \boldsymbol{\eta}_1, t_1), \mathcal{E}_2 = (\rho_2, \boldsymbol{\eta}_2, t_2)$. Let $\mathbf{r} \leftarrow \mathbb{Z}_q^n$ be a uniformly random vector and $\tilde{\mathbf{e}}_1 \in \chi^{(d+2)m}$, $\tilde{\mathbf{e}}_2 \in \chi^m, \tilde{\mathbf{e}}_3 \in \chi^{3m}$ are error vectors. Let $\boldsymbol{\eta}_2 = (\eta'_1, \cdots, \eta'_d)$. Set $\mathbf{H}'_1 = \left[\mathbf{A}|\mathbf{A}_{t_2}|\eta'_1\mathbf{G}+\mathbf{B}_1|\cdots|\eta'_d\mathbf{G}+\mathbf{B}_d\right] \in \mathbb{Z}_q^{n\times(d+2)m}$ and $\mathbf{H}'_2 = \left[\mathbf{D}|\mathbf{D}_{\rho_2}|\mathbf{D}_{t_2}\right] \in \mathbb{Z}_q^{n\times 3m}$. The updated ciphertext $ct' = (\mathbf{c}', \mathbf{c}'_{out}, \tilde{\mathbf{c}}')$ has the following form:

$$\begin{aligned}
\mathbf{c}' &= [\mathbf{c}'_{in}, \bar{\mathbf{c}}'_1, \mathbf{c}'_1, \cdots, \mathbf{c}'_d] = \Delta_1^\top\cdot[(\mathbf{c}_{in}, \bar{\mathbf{c}}_1, \mathbf{c}_1, \cdots, \mathbf{c}_d)] + \mathbf{H}_1^{'\top}\mathbf{r} + \tilde{\mathbf{e}}_1 \\
&= \mathbf{H}_1^{'\top}(\mathbf{s}+\mathbf{r}) + \underbrace{\Delta_1^\top\mathbf{e}+\tilde{\mathbf{e}}_1}. \quad \left(\text{Parse } \mathbf{p} \text{ as } [e'_{in}|\mathbf{e}'_0|\mathbf{e}'_{f_{\eta_1}}|\cdots|\mathbf{e}'_{f_{\eta_\ell}}]\right) \\
\mathbf{c}'_{out} &= \mathbf{c}_{out} + \mathbf{U}^\top\mathbf{r} + \tilde{\mathbf{e}}_2^{\mathbf{p}} = \mathbf{U}^\top(\mathbf{s}+\mathbf{r}) + \mathbf{e}_{out} + \tilde{\mathbf{e}}_2 + \lfloor q/2\rfloor\cdot\mathbf{m}. \\
\tilde{\mathbf{c}}' &= [\tilde{\mathbf{c}}'_0|\tilde{\mathbf{c}}'_1|\tilde{\mathbf{c}}'_2] = \Delta_2^\top\cdot[\mathbf{c}|\tilde{\mathbf{c}}] + \mathbf{H}_2^{'\top}\mathbf{r} + \tilde{\mathbf{e}}_3 \ = \mathbf{H}_2^{'\top}(\mathbf{s}+\mathbf{r}) + \Delta_2^\top\begin{bmatrix}\mathbf{e}\\ \mathbf{e}'\end{bmatrix} + \tilde{\mathbf{e}}_3.
\end{aligned}$$

If ρ_2 is not revoked at time t', then there are $(\mathbf{E}'_{1,\theta}, \mathbf{E}'_{2,\theta})$, which satisfies $[\mathbf{D}|\mathbf{D}_{\rho_2}]\cdot\mathbf{E}'_{1,\theta} + [\mathbf{D}|\mathbf{D}_{t'}]\cdot\mathbf{E}'_{2,\theta} = \mathbf{A}_{\rho_2}$. For $\tilde{\mathbf{c}}' = [\tilde{\mathbf{c}}'_0|\tilde{\mathbf{c}}'_1|\tilde{\mathbf{c}}'_2]$, it holds $\bar{\mathbf{c}}'_0 = \mathbf{E}'^\top_{1,\theta}[\tilde{\mathbf{c}}'_0|\tilde{\mathbf{c}}'_1] + \mathbf{E}'^\top_{2,\theta}[\tilde{\mathbf{c}}'_0|\tilde{\mathbf{c}}'_2]$. Parse $\mathbf{E}'_{i,\theta} = (\mathbf{E}_{i,\theta}^{'(0)}, \mathbf{E}_{i,\theta}^{'(1)})$ for $i \in \{1,2\}$. We can form a matrix $\mathbf{E} = \begin{bmatrix}\mathbf{E}_{1,\theta}^{'(0)} + \mathbf{E}_{2,\theta}^{'(0)}\\ \mathbf{E}_{1,\theta}^{'(1)}\\ \mathbf{E}_{2,\theta}^{'(1)}\end{bmatrix} \in \mathbb{Z}^{3m\times m}$, which satisfies $\bar{\mathbf{c}}'_0 = \mathbf{E}^\top\cdot\tilde{\mathbf{c}}'$.

$$\begin{aligned}
\text{We have } \bar{\mathbf{c}}'_0 &= \mathbf{E}^\top\cdot\tilde{\mathbf{c}}' = \mathbf{E}^\top\cdot\left([\mathbf{D}|\mathbf{D}_{\rho_2}|\mathbf{D}_{t'}]^\top(\mathbf{s}+\mathbf{r}) + \Delta_2^\top\begin{bmatrix}\mathbf{e}\\ \mathbf{e}'\end{bmatrix} + \tilde{\mathbf{e}}_3\right) \\
&= \mathbf{A}_{\rho_2}^\top(\mathbf{s}+\mathbf{r}) + \underbrace{\mathbf{E}^\top\cdot\left(\Delta_2^\top\begin{bmatrix}\mathbf{e}\\ \mathbf{e}'\end{bmatrix} + \tilde{\mathbf{e}}_3\right)}_{\mathsf{error}'_1}.
\end{aligned}$$

After evaluating $\mathbf{c}'_{f_{\eta_j}} \leftarrow \mathsf{Eval}_{ct}(\{\eta_i, \mathbf{B}_i, \mathbf{c}'_i\}_{i=1}^d, f_{\eta_j})$ for all $j \in \{1, \cdots, \ell\}$, we have the magnitude of the noise in $\mathbf{c}'_{f_{\eta_j}}$, i.e., $\left\|\mathbf{e}'_{f_{\eta_j}}\right\| < \Delta$. Therefore, during decryption of updated ciphertexts, one obtains:

$$\begin{aligned}
\mathbf{m} &= \mathbf{c}'_{out} - \mathbf{R}'^\top\left[\mathbf{c}'_{in}\middle|\bar{\mathbf{c}}'_0\middle|\bar{\mathbf{c}}'_1\middle|\mathbf{c}'_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{c}'_{f_{\eta_\ell}}\right] \\
&= \mathbf{c}'_{out} - \mathbf{R}'^\top\left(\left[\mathbf{A}_0|\mathbf{A}_{\rho_2}|\mathbf{A}_{t'}|\cdots|\mathbf{B}_{f_{\eta_\ell}}\right]^\top(\mathbf{s}+\mathbf{r}) + \left[\mathbf{e}'_{in}|\ \ \mathsf{error}'_1\ |\mathbf{e}'_0|\cdots|\mathbf{e}'_{f_{\eta_\ell}}\right]\right) \\
&= \lfloor q/2\rfloor\cdot\mathbf{m} + \underbrace{\mathbf{e}_{out} + \tilde{\mathbf{e}}_3 - \mathbf{R}'^\top\left[\mathbf{e}'_{in}|\ \ \mathsf{error}'_1\ |\mathbf{e}'_0|\cdots|\mathbf{e}'_{f_{\eta_\ell}}\right]}_{\mathsf{error}'_2}.
\end{aligned}$$

Note that the tokens Δ_1, Δ_2 are formed with the matrices sampled via the SampleRight algorithm, matrices chosen from χ, the identity matrix, and the zero matrix. The structure depends on the number of changes in tags required. To find the norm of Δ_1 and Δ_2, we consider the form for which the norm will be maximum from all the possible cases. Following Lemma 1 and the property of χ-distribution, we have $\left\|\Delta_1^\top\right\|_2 < (d+1)m\sigma + \chi_{max}$ and $\left\|\Delta_2^\top\right\|_2 < (d+3)m\sigma + \chi_{max}$. Since $\mathbf{E}'_{1,\theta}, \mathbf{E}'_{2,\theta} \in \mathbb{Z}^{2m\times m}$ are sampled from the distribution $\mathcal{D}_\sigma\big(\Lambda_q^{(\mathbf{A}_{\rho_2}-\mathbf{V}_\theta)}(\mathbf{D}|\mathbf{D}_{\rho_2})\big)$, $\mathcal{D}_\sigma\big(\Lambda_q^{\mathbf{V}_\theta}(\mathbf{D}|\mathbf{D}_{t_2})\big)$, respectively, we know that $[\mathbf{D}|\mathbf{D}_{\rho_2}]\cdot\mathbf{E}_{1,\theta} = \mathbf{A}_{\rho_2} - \mathbf{V}_\theta$, $[\mathbf{D}|\mathbf{D}_{t_2}]\cdot\mathbf{E}_{2,\theta} = \mathbf{V}_\theta$, and by Lemma 1, $\left\|\mathbf{E}^\top\right\|_2 < 3m\sigma$ with overwhelming probability. Since, $\tilde{e}_3 \in \chi^{3m}$, we know $\|\tilde{e}_3\| \leq \chi_{max}$. We have $\mathbf{e} = \left[\mathbf{I}_m|\mathbf{R}_0|\mathbf{R}_1|\cdots|\mathbf{R}_d\right]^\top\cdot\mathbf{e}_0 \in \mathbb{Z}_q^{(d+2)m}$ and $\mathbf{e}' = \left[\mathbf{I}_m|\mathbf{S}_1|\mathbf{S}_2\right]^\top\cdot\mathbf{e}'_0 \in \mathbb{Z}_q^{3m}$, where $\mathbf{R}_0, \mathbf{R}_1, \cdots, \mathbf{R}_d, \mathbf{S}_1, \mathbf{S}_2$ are random in $\{+1,-1\}^{m\times m}$, by Lemma 1, $\left\|\mathbf{R}_i^\top\right\|_2 < 20\sqrt{m}$ for $i \in \{0,1,\cdots,d\}$ and $\left\|\mathbf{S}_1^\top\right\|_2, \left\|\mathbf{S}_2^\top\right\|_2 < 20\sqrt{m}$ with overwhelming probability. Also, $\mathbf{e}_0, \mathbf{e}'_0 \in \chi^m$, we know $\|\mathbf{e}_0\|, \|\mathbf{e}'_0\| \leq \chi_{max}$. We have $\left\|\begin{bmatrix}\mathbf{e}\\ \mathbf{e}'\end{bmatrix}\right\| < 2\chi_{max} + 20\sqrt{m}(d+3)\chi_{max}$. So, we have $\|\mathsf{error}'_1\| = \left\|\mathbf{E}^\top\cdot\left(\Delta_2^\top\begin{bmatrix}\mathbf{e}\\ \mathbf{e}'\end{bmatrix} + \tilde{\mathbf{e}}_3\right)\right\| < 20\sqrt{m}((d+3)m\sigma\chi_{max})^2$ with overwhelming probability.

Since, $\mathbf{e}_{out} \in \chi^m$, we know $\|\mathbf{e}_{out}\| \leq \chi_{max}$. Since $\mathbf{R}' \in \mathbb{Z}^{(\ell+3)m\times n}$ is sampled from the distribution $\mathcal{D}_{\sigma_{\ell+1}}\big(\Lambda_q^{\mathbf{U}}(\mathbf{A}|\mathbf{A}_{\rho_2}|\mathbf{A}_{t_2}|\mathbf{B}_{f_{\eta'_1}}|\cdots|\mathbf{B}_{f_{\eta'_\ell}})\big)$, we know that $[\mathbf{A}|\mathbf{A}_{\rho_2}|\mathbf{A}_{t_2}|\mathbf{B}_{f_{\eta'_1}}|\cdots|\mathbf{B}_{f_{\eta'_\ell}}]\cdot\mathbf{R}' = \mathbf{U}$ and by Lemma 1, $\left\|\mathbf{R}'^\top\right\| < (\ell+3)m\sigma_{\ell+1}$ with overwhelming probability. So we have $\|\mathsf{error}'_2\| \leq 3\alpha_{\mathcal{F}}^3\cdot\chi_{max}^2\cdot(d+3)^2(\ell+3)^2\cdot m^{\frac{\ell+1}{2}+3}$ with overwhelming probability. Let $\beta_2 = 3\alpha_{\mathcal{F}}^3\cdot\chi_{max}^2\cdot(d+3)^2(\ell+3)^2\cdot m^{\frac{\ell+1}{2}+3}$. By choosing the parameters such that $\beta_2 < q/4$, the decryption of the updated ciphertext will be correct. For the correctness of the scheme, choose $\beta = max\{\beta_1, \beta_2\}$ and set the parameters such that, $\beta < q/4$, i.e., $3\alpha_{\mathcal{F}}^3\cdot\chi_{max}^2\cdot(d+3)^2(\ell+3)^2\cdot m^{\frac{\ell+1}{2}+3} < q/4$.

□

C Proof of Theorem 2

Proof. We build an dLWE algorithm $\mathcal{B}$ that uses a selective DPE attacker $\mathcal{A}$ to solve dLWE. We will demonstrate that if there is a PPT adversary $\mathcal{A}$ succeeding in breaking the IND-sDPE-CPA security of our DPE scheme, then we can use it to construct a PPT algorithm $\mathcal{B}$ to solve $\mathsf{dLWE}_{n,3m,q,\chi}$. $\mathcal{A}$ sends the target positive tag $\rho^* \in \mathcal{P}$, the target d-tuple of negative tags $\boldsymbol{\eta}^* = (\eta_1^*, \cdots, \eta_d^*) \in \mathcal{N}^d$, and time $t^* \in \mathcal{T}$ to the challenger. Let P^* be the set of revoked positive tags at t^*. The challenger initializes three empty sets $P, C, \overline{C}$ to record puncture key queries, corrupt key queries, and token queries. Also, the challenger introduces a counter numCT to 0, a key-value store $\mathcal{H}$ to be empty, and a set Derive to be empty. It sets a binary tree BT with at least N leaf nodes. The proof proceeds in a sequence of games. The first game is identical to the original IND-sDPE-CPA

game from the definition 3. The last two games are indistinguishable due to the hardness of the dLWE problem.

Game 0: This is the original IND-sDPE-CPA game from definition between an adversary $\mathcal{A}$ against scheme and an IND-sDPE-CPA challenger.

Game 1: This game is same as Game 0, except the generation of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d, \mathbf{D}_0, \mathbf{D}_1$ in the public key pk during Set up phase. The challenger chooses $(d+4)$ uniformly random matrices $\bar{\mathbf{R}}_0^*, \mathbf{R}_0^*, \mathbf{R}_1^*, \cdots, \mathbf{R}_d^*$ and $\mathbf{S}_1^{'*}, \mathbf{S}_2^{'*}$ from $\{+1, -1\}^{m \times m}$. Set $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d$ and $\mathbf{D}_0, \mathbf{D}_1$ as follows:

$\mathbf{A}_0 = \mathbf{A}\bar{\mathbf{R}}_0^*$, $\mathbf{A}_1 = \mathbf{A}\mathbf{R}_0^* - \mathbf{H}_{t^*}\mathbf{G}$, $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i^* - \eta_i^*\mathbf{G}$ for $i \in \{1, \cdots, d\}$,
$\mathbf{D}_0 = \mathbf{D}\mathbf{S}_1^{'*} - \mathbf{H}_{\rho^*}\mathbf{G}$, and $\mathbf{D}_1 = \mathbf{D}\mathbf{S}_2^{'*} - \mathbf{H}_{t^*}\mathbf{G}$.

The challenger generate $\mathbf{A}, \mathbf{D}, \mathbf{U}, \mathbf{G}$ same as Game 0, and sends $\mathsf{pk} = \{\mathbf{A}, \mathbf{D}, \mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d, \mathbf{D}_0, \mathbf{D}_1, \mathbf{U}, \mathbf{G}\}$ to the adversary $\mathcal{A}$. The remainder of the game is same as Game 0. Due to lemma 8 (Left-over Hash Lemma), $\mathbf{A}\bar{\mathbf{R}}_0^*, \mathbf{A}\mathbf{R}_0^*$ and $\mathbf{A}\mathbf{R}_1^*, \cdots, \mathbf{A}\mathbf{R}_d^*$, and $\mathbf{D}\mathbf{S}_1^{'*}, \mathbf{D}\mathbf{S}_2^{'*}$ are statistically indistinguishable with uniform distribution. So, $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d$, $\mathbf{D}_0, \mathbf{D}_1$ as defined above, are close to uniform. Hence, Game 0 and Game 1 are statistically indistinguishable.

Game 2: Here, the challenger chooses random $\mathbf{A}, \mathbf{D}$ from $\mathbb{Z}_q^{n \times m}$ instead of having from TrapGen algorithm and the construction of $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}_1, \cdots, \mathbf{B}_d, \mathbf{D}_0, \mathbf{D}_1$, $\mathbf{U}, \mathbf{G}$ in pk remain same as Game 1. The adversary issues following queries adaptively and the challenger does as follows:

$\mathcal{Q}_{Puncture}(\rho, \eta)$: Here, the challenger first defines $\mathbf{V}_\theta$ for each $\theta \in \mathsf{BT}$ as follows:

- If $\theta \in Path(\rho^*)$, pick $\mathbf{E}_{1,\theta}^* \leftarrow \mathcal{D}_{\mathbb{Z}^{2m \times m}, s}$ and $\mathbf{V}_\theta = \mathbf{A}_{\rho^*} - [\mathbf{D}|\mathbf{D}_{\rho^*}] \cdot \mathbf{E}_{\mathbf{1},\theta}^*$.
- If $\theta \notin Path(\rho^*)$, pick $\mathbf{E}_{2,\theta}^* \leftarrow \mathcal{D}_{\mathbb{Z}^{2m \times m}, s}$ and $\mathbf{V}_\theta = [\mathbf{D}|\mathbf{D}_{\mathbf{t}^*}] \cdot \mathbf{E}_{\mathbf{2},\theta}^*$.

Given a positive tag ρ and a negative tag η, consider following two cases:

1. *Query for $\rho(\neq \rho^*)$*:
   - When $\mathcal{A}$ issues the first puncture query under ρ, the challenger compute $\mathsf{sk}_{\mathsf{N}_\rho}^\rho$, where $\mathsf{N}_\rho = \{\eta\}$ as follows:
     Construct $\mathbf{A}_\rho = \mathbf{A}_0 + \mathbf{H}_\rho\mathbf{G} = \mathbf{A}\bar{\mathbf{R}}_0^* + (\mathbf{H}_\rho - \mathbf{H}_{\rho^*})\mathbf{G}$. Since, $\mathbf{T}_\mathbf{G}$ is a trapdoor for $\mathbf{G}$, so it's also a trapdoor for $(\mathbf{H}_\rho - \mathbf{H}_{\rho^*})\mathbf{G}$, as $(\mathbf{H}_\rho - \mathbf{H}_{\rho^*}) \neq 0$ by definition of FRD. Then obtain $\mathbf{T}_{(\mathbf{A}|\mathbf{A}_\rho)}^{\mathrm{EL}} \leftarrow \mathsf{ExtendLeft}(\mathbf{A}, (\mathbf{H}_\rho - \mathbf{H}_{\rho^*})\mathbf{G}, \mathbf{T}_\mathbf{G}, \bar{\mathbf{R}}_0^*)$. Evaluate $\mathbf{B}_{f_\eta} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\{\mathbf{B}_i\}_{i=1}^d, f_\eta)$. Compute $\mathbf{T}_{\{\eta\}}^{\rho,\mathrm{ER}} \leftarrow \mathsf{ExtendRight}\left(\left[\mathbf{A}\middle|\mathbf{A}_\rho\middle|\mathbf{B}_{f_\eta}\right], \mathbf{T}_{(\mathbf{A}|\mathbf{A}_\rho)}^{\mathrm{EL}}, \mathbf{B}_{f_\eta}\right)$. Finally, compute a randomized trapdoor $\mathbf{T}_{\{\eta\}}^\rho \leftarrow \mathsf{RandBasis}((\mathbf{A}|\mathbf{A}_\rho|\mathbf{B}_{f_\eta}), \mathbf{T}_{\{\eta\}}^{\rho,\mathrm{ER}}, \sigma_0)$. Set this as the punctured secret key $\mathsf{sk}_{1,\mathsf{N}_\rho}^\rho = \mathbf{T}_{(\mathbf{A}|\mathbf{A}_\rho)}$ for positive tag ρ, where $\mathsf{N}_\rho = \{\eta\}$, the set of containing negative tags.
   - To compute sk_2^ρ, the challenger proceeds as follows:
     Construct $\mathbf{D}_\rho = \mathbf{D} + \mathbf{H}_\rho\mathbf{G} = \mathbf{D}\mathbf{S}_1^{'*} + (\mathbf{H}_\rho - \mathbf{H}_{\rho^*})\mathbf{G}$. Since, $T_\mathbf{G}$ is a trapdoor for $\mathbf{G}$, so it's also a trapdoor for $(\mathbf{H}_\rho - \mathbf{H}_{\rho^*})\mathbf{G}$, as $(\mathbf{H}_\rho - \mathbf{H}_{\rho^*}) \neq 0$ by definition of FRD.
     Then sample $\mathbf{E}_{1,\theta} \leftarrow \mathsf{SampleLeft}\left(\mathbf{D}, \mathbf{S}_1^{'*}, (\mathbf{H}_\rho - \mathbf{H}_{\rho^*}), \mathbf{A}_\rho - \mathbf{V}_\theta, \sigma\right)$ for each $\theta \in Path(\rho)$. Note that $\mathbf{E}_{1,\theta} \in \mathbb{Z}^{2m \times m}$ and $[\mathbf{D}|\mathbf{D}_\rho] \cdot \mathbf{E}_{1,\theta} = \mathbf{A}_\rho - \mathbf{V}_\theta$. Set $\mathsf{sk}_2^\rho = (\theta, \mathbf{E}_{1,\theta})_{\theta \in Path(\rho)}$.

– Set $\mathsf{sk}^{\rho}_{\mathsf{N}_\rho} = \{\mathsf{sk}^{\rho}_{1,\mathsf{N}_\rho}, \mathsf{sk}^{\rho}_2\}$, where $\mathsf{N}_\rho = \{\eta\}$ is the set of containing negative tags and add $(\rho, \mathsf{sk}^{\rho}_{\mathsf{N}_\rho}, \mathsf{N}_\rho)$ to P. For further queries on ρ, use Punc algorithm accordingly and update the N_ρ and $\mathsf{sk}^{\rho}_{1,\mathsf{N}_\rho}$ in the tuple $(\rho, \mathsf{sk}^{\rho}_{\mathsf{N}_\rho}, \mathsf{N}_\rho)$.

2. *Query for* $\rho = \rho^*$:
   – The challenger add η to T_{ρ^*}, set $\mathsf{sk}^{\rho^*}_2 = (\theta, \mathbf{E}^*_{1,\theta})_{\theta \in Path(\rho^*)}$. Observe that $\mathbf{E}^*_{1,\theta} \in \mathbb{Z}^{2m \times m}$ and $[\mathbf{D}|\mathbf{D}_{\rho^*}] \cdot \mathbf{E}^*_{1,\theta} = \mathbf{A}_{\rho^*} - \mathbf{V}_\theta$. Set $\mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = \{-, \mathsf{sk}^{\rho^*}_2\}$. Construct a tuple $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}, \mathsf{N}_{\rho^*})$ and add it to P. For further puncture queries on ρ^*, update the N_{ρ^*} in P.

$\mathcal{Q}_{Corrupt}(\rho)$: The first time $\mathcal{A}$ makes a corruption query for ρ, challenger responses as the follows, and for all subsequent queries for ρ, returns $\perp$.

1. *Query for* ρ $(\neq \rho^*)$: Based on the restrictions, defined in 3, it returns either $\perp$ or the most recent key $\mathsf{sk}^{\rho}_{\mathsf{N}_\rho}$ from the table P. Add $(\rho, \perp/\mathsf{sk}^{\rho}_{\mathsf{N}_\rho}, \mathsf{N}_\rho)$ to C.
2. *Query for* ρ^*: For the tuple $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = \{-, \mathsf{sk}^{\rho^*}_2\}, \mathsf{N}_{\rho^*})$ in P, where N_{ρ^*} is the set $\{\eta_1, \cdots, \eta_k\}$, evaluate $\mathbf{B}_{f_{\eta_j}} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\{\mathbf{B}_i\}_{i=1}^d, f_{\eta_j})$, $\forall\ j \in \{1, \cdots, k\}$. Let $\mathsf{L} = [\mathbf{A}|\mathbf{A}_{\rho^*}|\mathbf{B}_{f_{\eta_1}}|\cdots|\mathbf{B}_{f_{\eta_k}}]$. To compute $\mathsf{sk}^{\rho^*}_{1,\mathsf{N}_{\rho^*}}$, it proceeds as follows:
   – If $\mathsf{N}^*_\rho \cap \{\eta^*_1, \cdots, \eta^*_d\} = \emptyset$: Observe that $[\mathbf{A}|\mathbf{A}_{\rho^*}] = [\mathbf{A}|\mathbf{A}\bar{\mathbf{R}}^*_{\mathbf{0}} + \mathbf{H}_{\rho^*}\mathbf{G}]$. Since, $T_{\mathbf{G}}$ is a trapdoor for $\mathbf{G}$, so it's also a trapdoor for $\mathbf{H}_{\rho^*}\mathbf{G}$, as $\mathbf{H}_{\rho^*} \neq 0$. Compute $\mathbf{T}^{\mathrm{EL}}_{[\mathbf{A}|\mathbf{A}_{\rho^*}]} \leftarrow \mathsf{ExtendLeft}(\mathbf{A}, \mathbf{H}_{\rho^*}\mathbf{G}, \mathbf{T}_{\mathbf{G}}, \bar{\mathbf{R}}^*_0)$.
   Compute $\mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}} \leftarrow \mathsf{ExtendRight}\left([\mathbf{A}|\mathbf{A}_{\rho^*}], \mathbf{T}^{\mathrm{EL}}_{[\mathbf{A}|\mathbf{A}_{\rho^*}]}, \left[\mathbf{B}_{f_{\eta_1}}\middle|\cdots\middle|\mathbf{B}_{f_{\eta_k}}\right]\right)$.
   Observe that $\mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}}$ is a trapdoor for L. Compute a randomized trapdoor $\mathbf{T}^{\rho^*}_{\mathsf{N}_{\rho^*}} \leftarrow \mathsf{RandBasis}(\mathsf{L}, \mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}}, \sigma_0)$. Set $\mathsf{sk}^{\rho^*}_{1,\mathsf{N}_{\rho^*}} = \mathbf{T}^{\rho^*}_{\mathsf{N}_{\rho^*}}$. Update the tuple from $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = \{-, \mathsf{sk}^{\rho^*}_2\}, \mathsf{N}_{\rho^*})$ to $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = \{\mathsf{sk}^{\rho^*}_{1,\mathsf{N}_{\rho^*}}, \mathsf{sk}^{\rho^*}_2\}, \mathsf{N}_{\rho^*})$.
   – If $\mathsf{N}^*_\rho \cap \{\eta^*_1, \cdots, \eta^*_d\} \neq \emptyset$: It implies that there exist at least one $\eta_i \in \mathsf{N}^*_\rho$ such that $f_{\eta_i}(\boldsymbol{\eta}^*) \neq 0$. Without loss of generality, we assume that $f_{\eta_k}(\boldsymbol{\eta}^*) \neq 0$. Compute $\mathbf{S}^*_{f_{\eta_k}} \leftarrow \mathsf{Eval}_{sim}(f_{\eta_k}, \{\eta^*_j, \mathbf{R}^*_j\}_{j=1}^d, \mathbf{A})$, and let $\mathbf{B}_{f_{\eta_k}} = \mathbf{A}\mathbf{S}^*_{f_{\eta_k}} - f_{\eta_k}(\boldsymbol{\eta}^*)\mathbf{G}$, where, $\left\|\mathbf{S}^*_{f_{\eta_k}}\right\|_2 \leq \alpha_{\mathcal{F}}$.
   Compute $\mathbf{T}^{\mathrm{EL}}_{[\mathbf{A}|\mathbf{B}_{f_{\eta_k}}]} \leftarrow \mathsf{ExtendLeft}\left(\mathbf{A}, f_{\eta_k}(\boldsymbol{\eta}^*)\mathbf{G}, \mathbf{T}_{\mathbf{G}}, \mathbf{S}^*_{f_{\eta_k}}\right)$. Compute a trapdoor $\mathbf{T}^{\mathrm{ER}}_{[\mathbf{A}|\mathbf{B}_{f_{\eta_k}}|\mathbf{A}_{\rho^*}|\mathbf{B}_{f_{\eta_1}}|\cdots|\mathbf{B}_{f_{\eta_{k-1}}}]}$ by running $\mathsf{ExtendRight}\left([\mathbf{A}|\mathbf{B}_{f_{\eta_k}}], \mathbf{T}^{\mathrm{EL}}_{(\mathbf{A}|\mathbf{B}_{f_{\eta_k}})}, (\mathbf{A}_{\rho^*}|\mathbf{B}_{f_{\eta_1}}|\cdots|\mathbf{B}_{f_{\eta_{k-1}}})\right)$. By switching the rows of the matrix $\mathbf{T}^{\mathrm{ER}}_{(\mathbf{A}|\mathbf{B}_{f_{\eta_k}}|\mathbf{A}_{\rho^*}|\mathbf{B}_{f_{\eta_1}}|\cdots|\mathbf{B}_{f_{\eta_{k-1}}})}$, we get a trapdoor $\mathbf{T}^{\mathrm{ER}}_{\mathsf{L}}$ for L. Set $\mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}} = \mathbf{T}^{\mathrm{ER}}_{\mathsf{L}}$ and by lemma 2, $\left\|\mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}}\right\|_{\mathrm{GS}} \leq \sqrt{5}\alpha_{\mathcal{F}}$.
   Compute a randomized trapdoor $\mathbf{T}^{\rho^*}_{\mathsf{N}_{\rho^*}} \leftarrow \mathsf{RandBasis}(\mathsf{L}, \mathbf{T}^{\rho^*,\mathrm{ER}}_{\mathsf{N}_{\rho^*}}, \sigma_0)$.
   Set $\mathsf{sk}^{\rho^*}_{1,\mathsf{N}_{\rho^*}} = \mathbf{T}^{\rho^*}_{\mathsf{N}_{\rho^*}}$ and update the tuple $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = (-, \mathsf{sk}^{\rho^*}_2), \mathsf{N}_{\rho^*})$ to $(\rho^*, \mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}} = \{\mathsf{sk}^{\rho^*}_{1,\mathsf{N}_{\rho^*}}, \mathsf{sk}^{\rho^*}_2\}, \mathsf{N}_{\rho^*})$ in P.
   Based on the restrictions, defined in 3, it returns either $\perp$ or the secret key $\mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}$ from P. Add the tuple $(\rho^*, \perp/\mathsf{sk}^{\rho^*}_{\mathsf{N}_{\rho^*}}, \mathsf{N}_{\rho^*})$ to C accordingly.

Observe that in the real scheme, the key $\mathsf{sk}^{\rho}_{1,\mathsf{N}_\rho}$ are generated using the algorithm $\mathsf{ExtendRight}, \mathsf{RandBasis}$ for every ρ. In contrast, during the simulation, they are drawn from $\mathsf{ExtendLeft}, \mathsf{ExtendRight}, \mathsf{RandBasis}$. Also, the matrices $\mathbf{E}_{1,\theta}$ are generated using $\mathsf{SampleRight}$ for every ρ in the real scheme. In contrast, during the simulation in , they are drawn from $\mathcal{D}_{\mathbb{Z}^{2m\times m},s}$ when $\rho = \rho^*$, and sampled using $\mathsf{SampleLeft}$ when $\rho \neq \rho^*$. The properties of these algorithms (see Section 3) ensure that the resulting distributions are statistically indistinguishable.

$\mathcal{Q}_{RevAccK}(t, \mathsf{P}_t)$: On input time t, the challenger first update the revoke list P_t based on the restriction provided in 3, and it returns $\mathsf{rk}^t_{\mathsf{P}_t}$ as follows.

1. *Query for* $t(\neq t^*)$: the challenger construct $\mathbf{D}_t = \mathbf{D} + \mathbf{H}_t\mathbf{G} = \mathbf{D}\mathbf{S}_1^{'*} + (\mathbf{H}_t - \mathbf{H}_{t^*})\mathbf{G}$. Since, $\mathbf{T}_\mathbf{G}$ is a trapdoor for $\mathbf{G}$, so it's also a trapdoor for $(\mathbf{H}_t - \mathbf{H}_{t^*})\mathbf{G}$, as $(\mathbf{H}_t - \mathbf{H}_{t^*}) \neq 0$ by definition of FRD.

   Then sample $\mathbf{E}_{2,\theta} \leftarrow \mathsf{SampleLeft}\left(\mathbf{D}, \mathbf{S}_1^{'*}, (\mathbf{H}_t - \mathbf{H}_{t^*}), \mathbf{V}_\theta, \sigma\right)$ for each $\theta \in \mathsf{KUNodes}(\mathsf{BT}, \mathsf{P}_t)$. Note that $\mathbf{E}_{2,\theta} \in \mathbb{Z}^{2m\times m}$ and $[\mathbf{D}|\mathbf{D}_t] \cdot \mathbf{E}_{2,\theta} = \mathbf{V}_\theta$.
   Set $\mathsf{rk}_t = (\theta, \mathbf{E}_{2,\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{P}_t)}$ and returns $\mathsf{rk}^t_{\mathsf{P}_t}$.
2. *Query for* $t = t^*$: for the challenge time period t^*, the challenger returns $(\theta, \mathbf{E}^*_{2,\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{P}_{t^*},t^*)}$. (Defined in $\mathcal{Q}_{Puncture}$). Note that $\mathbf{E}^*_{2,\theta} \in \mathbb{Z}^{2m\times m}$ and $[\mathbf{D}|\mathbf{D}_{t^*}]\cdot\mathbf{E}^*_{2,\theta} = \mathbf{V}_\theta$. Set $\mathsf{rk}_{t^*} = (\theta, \mathbf{E}^*_{2,\theta})_{\theta\in\mathsf{KUNodes}(\mathsf{BT},\mathsf{P}_t)}$ and returns $\mathsf{rk}^t_{\mathsf{P}_t}$.

Observe that in the real scheme, the matrices $\mathbf{E}_{2,\theta}$ are generated using the algorithm $\mathsf{SampleRight}$ for every t. In contrast, during the simulation, they are drawn from $\mathcal{D}_{\mathbb{Z}^{2m\times m},s}$ when $t = t^*$, and sampled using $\mathsf{SampleLeft}$ when $t \neq t^*$. The properties of these sampling algorithms (see Section 3) ensure that the resulting distributions are statistically indistinguishable.

$\mathcal{Q}_{Token}(\rho_1, \boldsymbol{\eta}_1, t_1 \to \rho_2, \boldsymbol{\eta}_2, t_2)$: The challenger consider the following two cases:
If $(\rho_1, \boldsymbol{\eta}_1, t_1) = (\rho^*, \boldsymbol{\eta}^*, t^*)$, it proceeds as follows.

– If $\boldsymbol{\eta}_1 = \boldsymbol{\eta}_2$: the challenger uniformly samples $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}$ from $\mathbb{Z}^{m\times m} \times \mathbb{Z}^{dm\times m}$, and $\mathbf{Y} \leftarrow \chi^{m\times m}$, respectively. Set

$$\Delta_1 = \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{X}^{(0)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}^{(1)} & \mathbf{I}_{dm\times dm} \end{bmatrix} \in \mathbb{Z}^{(d+2)m\times(d+2)m}$$

  If $\boldsymbol{\eta}_1 \neq \boldsymbol{\eta}_2$: let $\boldsymbol{\eta}_1 = (\eta_1, \cdots, \eta_d), \boldsymbol{\eta}_2 = (\eta'_1, \cdots, \eta'_d)$. Without loss of generality, we assume that first $j(\leq d)$ components of $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$ are different. We have $\eta_1 \neq \eta'_1, \cdots, \eta_j \neq \eta'_j$, and $\eta_{j+1} = \eta'_{j+1}, \cdots, \eta_d = \eta'_d$. Choose $\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_j \leftarrow \chi^{m\times m}$, and uniformly samples
  $\mathbf{X}_0 = (\mathbf{X}_0^{(0)}, \mathbf{X}_0^{(j+1)}, \cdots, \mathbf{X}_0^{(d)}) \in \underbrace{\mathbb{Z}^{m\times m} \times \cdots \times \mathbb{Z}^{m\times m}}_{(d-j+1)\text{ times}}$, and
  $\mathbf{X}_i = (\mathbf{X}_i^{(0)}, \mathbf{X}_i^{(j+1)}, \cdots, \mathbf{X}_i^{(d)}) \in \underbrace{\mathbb{Z}^{m\times m} \times \cdots \times \mathbb{Z}^{m\times m}}_{(d-j+1)\text{ times}}$ for each $i = \{1, \cdots, j\}$.

Set

$$\Delta_1 = \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{X}_0^{(0)} & \mathbf{X}_1^{(0)} & \cdots & \mathbf{X}_j^{(0)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_0 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Y}_j & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_0^{(j+1)} & \mathbf{X}_1^{(j+1)} & \cdots & \mathbf{X}_j^{(j+1)} & \mathbf{I}_{m\times m} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{X}_0^{(d)} & \mathbf{X}_1^{(d)} & \cdots & \mathbf{X}_j^{(d)} & \mathbf{0} & \cdots & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}^{(d+2)m\times(d+2)m}.$$

– If $\rho_1 = \rho_2$: the challenger uniformly samples $(\mathbf{X}'^{(0)}, \mathbf{X}'^{(1)})$ from $\mathbb{Z}^{(d+2)m\times m} \times \mathbb{Z}^{m\times m}$ and $\mathbf{Y}' \leftarrow \chi^{m\times m}$, respectively.

  Set $\Delta_2 = \begin{bmatrix} \mathbf{0}_{(d+2)m\times m} & \mathbf{0}_{(d+2)m\times m} & \mathbf{X}'^{(0)} \\ \mathbf{I}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} & \mathbf{X}'^{(1)} \\ \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{Y}' \end{bmatrix} \in \mathbb{Z}^{(d+5)m\times 3m}.$

  If $\rho_1 \neq \rho_2$: the challenger uniformly samples $\mathbf{X}'_1, \mathbf{X}'_2$ from $\mathbb{Z}^{(d+2)m\times m}$, and $\mathbf{Y}_1, \mathbf{Y}_2 \leftarrow \chi^{m\times m}$.

  Set $\Delta_2 = \begin{bmatrix} \mathbf{0}_{(d+2)m\times m} & \mathbf{X}'_1 & \mathbf{X}'_2 \\ \mathbf{I}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{Y}'_1 & \mathbf{0}_{m\times m} \\ \mathbf{0}_{m\times m} & \mathbf{0}_{m\times m} & \mathbf{Y}'_2 \end{bmatrix} \in \mathbb{Z}^{(d+5)m\times 3m}.$

If $(\rho_1, \boldsymbol{\eta}_1, t_1) \neq (\rho^*, \boldsymbol{\eta}^*, t^*)$, then challenger can use $\mathbf{T_G}$ to compute the token. Send $\mathsf{tk}^{t_1\to t_2}_{\rho_1,\boldsymbol{\eta}_1\to\rho_2,\boldsymbol{\eta}_2} = \{\Delta_1, \Delta_2\}$ or $\perp$ to the adversary $\mathcal{A}$ following the restriction defined in 3. The challenger adds this the token to the set $\overline{C}$.

$\mathcal{Q}_{\mathsf{Enc}}(\rho, \boldsymbol{\eta}, \mathbf{m}, t)$: Given a positive tag ρ, a d-tuple of negative tags $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)$, and message $\mathbf{m}$, time t, outputs ciphertext ct by running $\mathsf{Encrypt}(\cdot)$. Increment $numCt$ and add ct to the set $\mathcal{H}$ with key $(\rho, \boldsymbol{\eta}, t, numCt)$.

$\mathcal{Q}_{\mathsf{Update}}$: Given $\rho', \boldsymbol{\eta}', t' (> t)$ and a key $(\rho, \boldsymbol{\eta}, t, numCt)$, where $k \leq numCt$, outputs $\perp$ if there is no value in $\mathcal{H}$ with key $(\rho, \boldsymbol{\eta}, t, k)$. Otherwise, let ct be that value in $\mathcal{H}$ and it computes the updated ciphertext ct' by proceeding as follows:

– If there is a token $\mathsf{tk}^{t\to t'}_{\rho,\boldsymbol{\eta}\to\rho',\boldsymbol{\eta}'}$ in the set $\overline{C}$, it computes the updated ciphertext ct' using this key by running the algorithm $\mathsf{UpdateCT}$. Sends ct' to $\mathcal{A}$.
– If there is no token $\mathsf{tk}^{t\to t'}_{\rho,\boldsymbol{\eta}\to\rho',\boldsymbol{\eta}'}$ in the set $\overline{C}$, it first computes the key as in $\mathcal{Q}_{Token}(\rho, \boldsymbol{\eta}, t \to \rho', \boldsymbol{\eta}', t')$. Then, it computes the updated ciphertext ct' using this key by running the algorithm $\mathsf{UpdateCT}$ and sends ct' to $\mathcal{A}$. If $\mathcal{Q}_{Token}$ returns $\perp$, then challenger also returns $\perp$ to $\mathcal{A}$.

Game 2 is otherwise same as Game 1. Since the Keys and responses to the queries are statistically close to those in Game 1, the adversary $\mathcal{A}$'s advantage in Game 2 is at most negligibly different from its advantage in Game 1.

Game 3: Game 3 is identical to Game 2 except that the challenge ciphertext $ct^* = (\mathbf{c}^*, \mathbf{c}^*_{out}, \tilde{\mathbf{c}}^*)$ chosen randomly from $\mathbb{Z}_q^{(d+6)m}$. We show that Game 2 and Game 3 are computationally indistinguishable for a PPT adversary, by giving a reduction from the dLWE problem.

Reduction from dLWE: Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing Game 2 and Game 3. Using $\mathcal{A}$, we construct a dLWE solver $\mathcal{B}$.

- dLWE **instance**: $\mathcal{B}$ begins by obtaining an dLWE challenge consisting of three random matrices $\mathbf{A}, \mathbf{D}, \mathbf{U} \in \mathbb{Z}_q^{n\times m}$ and three vectors $\mathbf{c}'_{in}, \mathbf{c}'_{out}, \tilde{\mathbf{c}}'_0 \in \mathbb{Z}_q^m$. Here, $\mathbf{c}'_{in},\ \mathbf{c}'_{out},\ \tilde{\mathbf{c}}'_0$ are either random in $\mathbb{Z}_q^m$ or $\mathbf{c}'_{in} = \mathbf{A}^\top\mathbf{s}+\mathbf{e}_0,\ \mathbf{c}'_{out} = \mathbf{U}^\top\mathbf{s}+\mathbf{e}_{out}$ and $\tilde{\mathbf{c}}'_0 = \mathbf{D}^\top\mathbf{s} + \mathbf{e}'_0$ for some random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and some small noise vectors $\mathbf{e}_0,\ \mathbf{e}_{out},\ \mathbf{e}'_0, \in \chi^m$. The goal of $\mathcal{B}$ is to distinguish these two cases with non-negligible advantage by using $\mathcal{A}$.
- **Initial**: $\mathcal{A}$ begins by announcing the target positive tag $\rho^* \in \mathcal{P}$, the target d-tuple of negative tags $\boldsymbol{\eta}^* = (\eta_1^*, \cdots, \eta_d^*) \in \mathcal{N}^d$, and time $t^* \in \mathcal{T}$.
- **SetUp**: $\mathcal{B}$ constructs the public key as in Game 2. Note that $\mathbf{A}_1 = \mathbf{A}\mathbf{R}_0^* - \mathbf{H}_{t^*}\mathbf{G}$, and $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i^* - \eta_i^*\mathbf{G}$ for $i \in \{1, \cdots, d\}$, and $\mathbf{D}_0 = \mathbf{D}\mathbf{S}_1^{'*} - \mathbf{H}_{\rho^*}\mathbf{G}, \mathbf{D}_1 = \mathbf{D}\mathbf{S}_2^{'*} - \mathbf{H}_{t^*}\mathbf{G}$, where $\mathbf{R}_i^*$ for $i \in \{0, 1, \cdots, d\}$ and $\mathbf{S}_1^{'*}, \mathbf{S}_2^{'*}$ are uniformly random matrices from $\{+1, -1\}^{m\times m}$. It gives pk to $\mathcal{A}$.
- **Query Phase 1**: $\mathcal{B}$ answers $\mathcal{A}$'s all key queries as in Game 2.
- **Challenge**: $\mathcal{A}$ sends two messages $\mathbf{m}_0, \mathbf{m}_1 \in \{0,1\}^m$ to $\mathcal{B}$. $\mathcal{B}$ chooses a random bit $\beta \in \{0,1\}$ and compute $ct^* = (\mathbf{c}^*, \mathbf{c}^*_{out}, \tilde{\mathbf{c}}^*) \in \mathbb{Z}_q^{(d+6)m}$, where $\mathbf{c}^* = [\mathbf{c}^*_{in}|\bar{\mathbf{c}}^*_1|\mathbf{c}^*_1|\cdots|\mathbf{c}^*_d]$ and $\tilde{\mathbf{c}}^* = [\tilde{\mathbf{c}}^*_0|\tilde{\mathbf{c}}^*_1|\tilde{\mathbf{c}}^*_2]$, as follows:
$$\begin{cases} \mathbf{c}^*_{in} = \mathbf{c}'_{in},\ \bar{\mathbf{c}}^*_1 = (\mathbf{R}^*_0)^\top \cdot \mathbf{c}^*_{in},\ \mathbf{c}^*_i = (\mathbf{R}^*_i)^\top \cdot \mathbf{c}^*_{in} \text{ for } i \in \{1, \cdots, d\}, \\ \mathbf{c}^*_{out} = \mathbf{c}'_{out} + \lfloor q/2 \rfloor \cdot \mathbf{m}_\beta \ \in \mathbb{Z}_q^m, \\ \tilde{\mathbf{c}}^*_0 = \tilde{\mathbf{c}}'_0,\ \tilde{\mathbf{c}}^*_1 = (\mathbf{S}_1^{'*})^\top \cdot \tilde{\mathbf{c}}'_0,\ \tilde{\mathbf{c}}^*_2 = (\mathbf{S}_2^{'*})^\top \cdot \tilde{\mathbf{c}}'_0. \end{cases}$$
$\mathcal{B}$ sends ct^* to $\mathcal{A}$ as the challenge ciphertext. Increment $numCt$ and add $numCt$ to the set $Derive$. Store ct^*_β to the set $\mathcal{H}$ with key $(\rho^*, \boldsymbol{\eta}^*, t^*, numCt)$.
  - Suppose $\mathbf{c}'_{in}, \mathbf{c}'_{out}, \tilde{\mathbf{c}}'_0$ are generated by dLWE i.e. $\mathbf{c}'_{in} = \mathbf{A}^\top\mathbf{s}+\mathbf{e}_0,\ \mathbf{c}'_{out} = \mathbf{U}^\top\mathbf{s} + \mathbf{e}_{out}$ and $\tilde{\mathbf{c}}'_0 = \mathbf{D}^\top\mathbf{s} + \mathbf{e}'_0$. From the Encrypt algorithm, we have,
$$\begin{aligned} \mathbf{c}^* &= \left[\mathbf{c}^*_{in}\middle|\bar{\mathbf{c}}^*_1\middle|\mathbf{c}^*_1\middle|\cdots\middle|\mathbf{c}^*_d\right] = \left[\mathbf{I}_m\middle|\mathbf{R}^*_0\middle|\mathbf{R}^*_1\middle|\cdots\middle|\mathbf{R}^*_d\right]^\top \cdot (\mathbf{A}^\top\mathbf{s} + \mathbf{e}_0) \\ &= \left[\mathbf{A}\middle|\mathbf{A}_{t^*}\middle|\mathbf{B}_1 + \eta^*_1\mathbf{G}\middle|\cdots\middle|\mathbf{B}_d + \eta^*_d\mathbf{G}\right]^\top\mathbf{s} + \left[\mathbf{I}_m\middle|\mathbf{R}^*_0\middle|\mathbf{R}^*_1\middle|\cdots\middle|\mathbf{R}^*_d\right]^\top \cdot \mathbf{e}_0. \\ \tilde{\mathbf{c}}^* &= [\tilde{\mathbf{c}}^*_0|\tilde{\mathbf{c}}^*_1|\tilde{\mathbf{c}}^*_2] = \left[\mathbf{I}_m\middle|\mathbf{S}_1^{'*}\middle|\mathbf{S}_2^{'*}\right]^\top \cdot (\mathbf{D}^\top\mathbf{s} + \mathbf{e}'_0) \\ &= \left[\mathbf{D}\middle|\mathbf{D}_{\rho^*}\middle|\mathbf{D}_{t^*}\right]^\top\mathbf{s} + \left[\mathbf{I}_m\middle|\mathbf{S}_1^{'*}\middle|\mathbf{S}_2^{'*}\right]^\top \cdot \mathbf{e}'_0. \end{aligned}$$
    It is easy to see that $\mathbf{c}^*, \tilde{\mathbf{c}}^*$ are computed as in Game 2.
    Also, $\mathbf{c}^*_{out} = \mathbf{U}^\top\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \cdot \mathbf{m}_\beta$. Then ct^* is a valid ciphertext of $\mathbf{m}_\beta$ with the positive tag ρ^*, negative tag set $\boldsymbol{\eta}^*$, and time t^*.
  - When $\mathbf{c}'_{in}, \mathbf{c}'_{out}, \tilde{\mathbf{c}}'_0$ are random in $\mathbb{Z}_q^m$, by left-over hash lemma, $\bar{\mathbf{c}}^*_1, \mathbf{c}^*_1, \cdots, \mathbf{c}^*_d, \tilde{\mathbf{c}}'_1, \tilde{\mathbf{c}}'_2$ are statistically indistinguishable with uniform. Also, $\mathbf{c}^*_{out}$ is uniform. So, $ct^* = (\mathbf{c}^*, \mathbf{c}^*_{out}, \tilde{\mathbf{c}}^*)$ are uniform in $\mathbb{Z}_q^{(d+6)m}$, as in Game 3.
- **Query Phase 2**: As in Game 2 except the following constraints for $\mathcal{Q}_{\mathsf{Update}}$ oracle: Outputs $\perp$ if $k \in Derive$.
- **Guess**: $\mathcal{A}$ guesses if it is interacting with a Game 2 or Game 3 challenger. $\mathcal{B}$ outputs $\mathcal{A}$'s guess as the answer to the dLWE challenge it is trying to solve.

Hence, $\mathcal{B}$'s advantage in solving dLWE is the same as $\mathcal{A}$'s advantage in distinguishing Game 2 and Game 3, as required. This completes the proof.

□